COMPUTER OPERATOR AND PROGRAMMING ASSISTANT

NSQF LEVEL - 4

1st Year (Volume II of II)

TRADE THEORY

SECTOR: IT & ITES

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 : Computer Operator and Programming Assistant 1st Year (Volume II of II) - Trade Theory - NSQF Level 4

First Edition : November 2018 Copies : 1,000
First Reprint : January 2019 Copies : 2,000

Rs. 245/-

Published by:

NATIONAL INSTRUCTIONAL MEDIA INSTITUTE
P. B. No.3142, CTI Campus, Guindy Industrial Estate,
Guindy, Chennai - 600 032.
Phone : 044 - 2250 0248, 2250 0657, 2250 2421
Fax : 91 - 44 - 2250 0791
email : chennai-nimi@nic.in, nimi_bsnl@dataone.in
Website: www.nimi.gov.in

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 **Computer Operator and Programming Assistant Trade Theory 1st Year (Volume II of II) in IT & ITES Sector .** The NSQF Level - 4 Trade Theory 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 - 4 trainees will also get the opportunities to promote life long learning and skill development. I have no doubt that with NSQF Level - 4 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

RAJESH AGGARWAL
Director General/ Addl. Secretary
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 Directorate General of Training, 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 4) 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.

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.

R. P. DHINGRA
EXECUTIVE DIRECTOR

Chennai - 600 032

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 **Computer Operator and Programming Assistant** under the IT & ITES Sector

MEDIA DEVELOPMENT COMMITTEE MEMBERS

Shri. M. K. Sunil	–	Training Officer NSTI, Chennai Tamil Nadu
Shri. C.Jeyaprakash	–	Technical Head Cadd Cae Industrial School, Dindigul Tamil Nadu
Shri. S.Venkadesh Babu	–	Chief Executive Officer, NSree Media Technologies, Dindigul Tamil Nadu
Smt. R. Jeyalakshmi	–	Assistant Training officer, Govt. ITI (W), Coimbatore Tamil Nadu
Shri. K. Kumaravel	–	Assistant Training Officer, Govt. ITI, Central Prison Campus, Trichy Tamil Nadu
Shri. J. Malarkodi	–	Assistant Training Officer, Govt. ITI., Salem Tamil Nadu
Shri. J. Herman	–	Assitant Manager, Co-ordinator, 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 THEORY

The manual of trade theory consists of theorectical information for the Second Semester 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 theortical aspects with the skill covered in each exercise to the extent possible. This co-relation is maintained to help the trainees to develop the perceptional 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 system lab. 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.

TRADE PRACTICAL

The trade practical manual is intented to be used in workshop . It consists of a series of practical exercies to be completed by the trainees during the Second Semester 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 the prescribed syllabus are covered.

The manual is divided into five modules to maintain completancy of learning process in a stipulated time basis.

The skill training in the computer lab is planned through a series of practical exercises centred around some practical project. However, there are few instance 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 traninee. However the development team accept that there if a scope for further improvement. NIMI, looks forward to the suggestions from the experienced training faculty for improving the manual.

CONTENTS

LEARNING / ASSESSABLE OUTCOME

On completion of this book you shall be able to

- **Develop web pages using JavaScript.**
- **Develop simple spread sheets by embedding VBA.**
- **Maintain accounts using Accounting Software**
- **Browse, select and transact using E-Commerce websites.**
- **Secure information from Internet by using Cyber Security concept.**

Week No.	Learning Outcome	Professional Skills (Trade Practical) With indicative hours	Professional Knowledge (Trade Theory) Introduction to JavaScript
27 - 33	Develop web pages using JavaScript.	**JavaScript & creating Web page** 94. Practice with basic elements of JavaScript. (12 Hrs) 95. Embed JavaScript in HTML to display information in web pages, documentation and formatting of HTML source code. (18 Hrs) 96. Use JavaScript Variables, Data types, Constants and Operators. (18 Hrs) 97. Use Control statements and Loops in JavaScript. (18 Hrs) 98. Practice with switch case, loop controls and Errors in JavaScript. (18 Hrs) 99. Practice with Arrays in JavaScript page. (12 Hrs) 100. Practice with functions in JavaScript web page. (18 Hrs) 101. Practice with String, Math and Date functions in JavaScript. (24 Hrs) 102. Use online tool or open source software to develop and edit web pages containing Titles, different font sizes and colours, frames, lists, tables, images, image map, controls, CSS, forms, hyperlinks etc., use web template to create a web page of various styles. (36 Hrs) 103. Develop a simple web project using HTML, JavaScript and host it in IIS and a registered domain. (36 Hrs)	• Introduction to Programming and Scripting Languages. • 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.
34 - 41	Develop simple spread sheets by embedding VBA.	**Programming with VBA** 104. Practice with basic functions of VBA Editor. (3 Hrs) 105. Use form controls like buttons, Check boxes, Labels, Combo Box, Group Box, List Box, Option Button, Scroll Bar and Spin button. (12 Hrs) 106. Modify object properties in V B A program. (6 Hrs) 107. Write simple programs involving VBA Data types, Variables, Operators and Constants. (18 Hrs) 108. Create Message boxes and I n p u t	**Introduction to VBA, Features and Applications.** • Introduction to VBA features and applications. • Properties, events and methods associated with the Button, Check Box, Label, Combo Box, Group Box, Option Button, List Box, Scroll Bar and Spin button controls. • VBA Data types, Variables and Constants.

		boxes in VBA. (6 Hrs) 109. Work with conditional statements like if, Else-if, and Select. (12 Hrs) 110. Practice with Loop, Loop Control and Case statements in VBA. (15 Hrs) 111. Create and Manipulate Arrays in VBA. (12 Hrs) 112. Practice with string variables in VBA programming. (12 Hrs) 113. Write programs involving Mathematical, Conversion, Date and String Functions in VBA. (18 Hrs) 114. Create Functions, Procedures, Passing Parameters and Using Returned Data. (12 Hrs) 115. Practice with built in functions in VBA programs. (12 Hrs) 116. Create and edit macros. (12 Hrs) 117. Write code to work with Excel in VBA forms. (12 Hrs) 118. Practice with methods and events in VBA Programming. (24 Hrs) 119. Debug, Step through code, Breakpoints, find and fix errors while debugging. (18 Hrs) 120. Develop a simple project involving MS excel and VBA. (36 Hrs)	• Operators in VBA and operator precedence. • Mathematical Expressions in VBA. • Introduction to Arrays in VBA. • Introduction to Strings in VBA. • Conditional processing in VBA, using the IF, Else-if, Select Case Statements. • Introduction to Loops in VBA. • VBA message boxes and input boxes. • Introduction to Creating functions and Procedures in VBA. • Using the built in functions. • Introduction to Object Oriented Programming Concepts. Concepts of Classes, Objects, Properties and Methods. • The user forms and control in Excel VBA. • Introduction to Debugging Techniques.
42 - 45	Maintain accounts using accounting software.	**Using Accounting Software** 121. Practice Basic accounting with tally interface. (12 Hrs) 122. Create Company, Account and Voucher entry in Tally. (12 Hrs) 123. Generate reports for Invoice, Bill, Profit & Loss account etc. (10 Hrs) 124. Perform Cost Centre & Cost Category management. (12 Hrs) 125. Create and manage budgeting systems. (12 Hrs) 126. Create Scenario and Variance Analysis. (8 Hrs) 127. Use Tally for Costing, Ratio Analysis, Cash flow and Funds flow statements. (12 Hrs) 128. Analyze and Manage Inventory control. (10 Hrs) 129. Perform Point of Sales and Taxation (VAT, Excise, Service Tax). (8 Hrs) 130. Perform System Administration and use other Utilities. (8 Hrs) 131. Create users, take Backup &	**Using Accounting Software** • Basics of Accounting, Golden Rules of Accounting, Voucher Entry, Ledger Posting, Final Accounts Preparation. • Cash Book. Ratio Analysis, Depreciation, Stock Management. • Analysis of VAT, Cash Flow, Fund Flow Accounting. • Introduction to Tally, features and Advantages. • Implementing accounts in Tally. • Double entry system of book keeping. • Budgeting Systems, Scenario management and Variance Analysis.

		Restore of Company. (8 Hrs) 132. Use Multilingual Functionality in Tally. (8 Hrs)	
46	Browse, select and transact using Ecommerce websites Secure information from Internet by using cyber security concept.	**E Commerce** 133. Browse E-commerce websites viz. ebay, Amazon, flipkart, OLX, quikr etc. and prepare comparative statement of the main features of these sites. (8 Hrs) 134. Upload products for selling in ECommerce Sites and make online purchase from E Commerce sites.(14 Hrs) 135. Manage security issues in ECommerce and payment operations. (8 Hrs)	**E Commerce Concepts** • Introduction to ECommerce advantages. • Building business on the net. • Payment and Order Processing, Authorization, Chargeback and other payment methods. • Security issues and payment gateways. and
47		**Cyber Security:** 136. Protect information, computers and networks from viruses, spyware and other malicious code. (3 Hrs) 137. Provide firewall security for Internet connection and Network System. (6 Hrs) 138. Protect the computer against various internet threats. (3 Hrs) 139. Make backup copies of important file, data and information. (3 Hrs) 140. Secure your Wi-Fi networks using password, WEP, WPA-PSK, WPA2-PSK, SSID, MAC address filtering. Create individual user accounts for each member. (9 Hrs) 141. Limit member access to data and information, and restrict authority to install unnecessary downloads. (6 Hrs)	**Cyber Security:** • 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 cyber crimes.
48 - 49	Industrial Visit/Project work (**1.** Create and host a web site of atleast 6 web pages using JavaScript containing interactive objects, functions etc. **OR** **2.** Create a project with Excel & VBA on Payroll Systems. **OR** **3.** Create a company in Tally and post vouchers in it for a financial period. Vouchers should contain purchase, sales with VAT, contra, payment , receipts, cost centre cost category etc.)		
50-51	Revision		
52	Examination		

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.

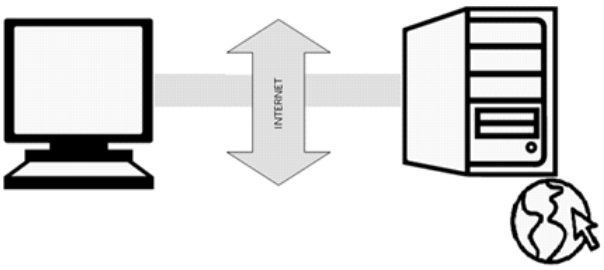
Disadvantages of scripts

- 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.

Fig 1



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 uses 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.

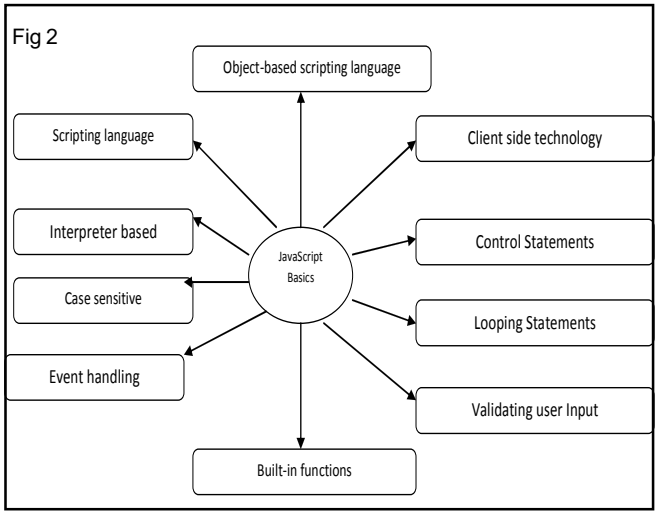
Sample JavaScript program

```
<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.

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.
- Instance 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.

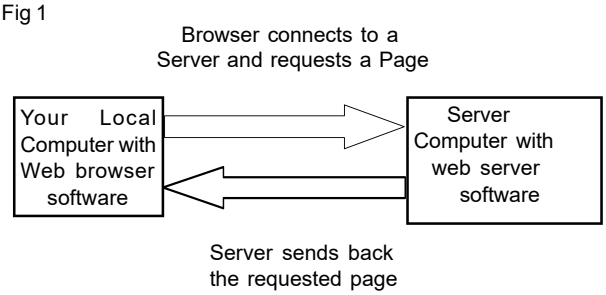
Introduction to Web servers and External JavaScript files

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)

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 an**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. Java Script 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.getElementById(“demo”).innerHTML = 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"
typeof0                    // Returns "number"
typeof81                   // Returns "number"
typeof8.14                 // 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",last Name;
                  "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        = 'Shanthi';
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.

00000000000000000000000000000101 (5)

11111111111111111111111111111010 (~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.

Operator	Function
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
in	Tests whether a value is found in an expression
instanceof	Tests whether an expression is an instance of an object

```
if(5<2) {
// do something
}
```

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 runnign 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*”);

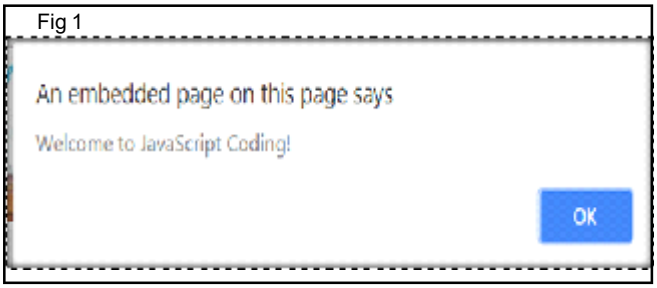
Note: The window.alert() method can be written without the window prefix.

Example 15

alert (“Welcome to Java Script Coding!;)

The result is shown in Fig 1.

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*");

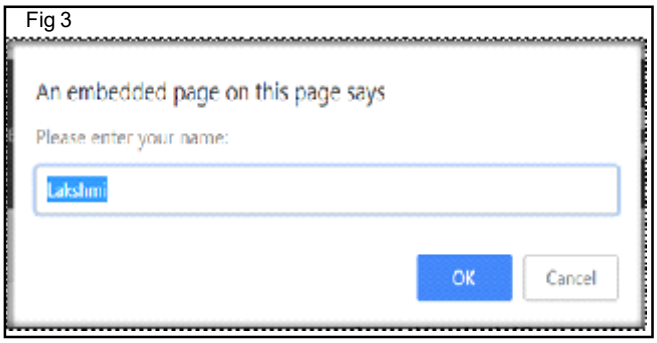
Note: The window.prompt() method can be written without the window prefix.

Example 17

```
var tname = promp t("Please Enter your Name", "Lakshmi");
    if (tname == null || tname == "")
      {txt = "User cancelled the prompt.";
    }
    else
       {txt = "Hello " + tname + "! Congratulations!!!!!");
    }
```

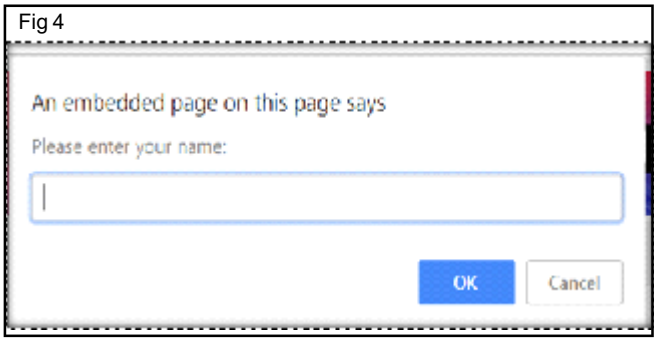
The result is is shown in Fig 3.

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 whenclick the Cancel button even when if the box has text 'Lakshmi' it gives the message "User cancelled the Prompt".

Fig 4



Line Breaks

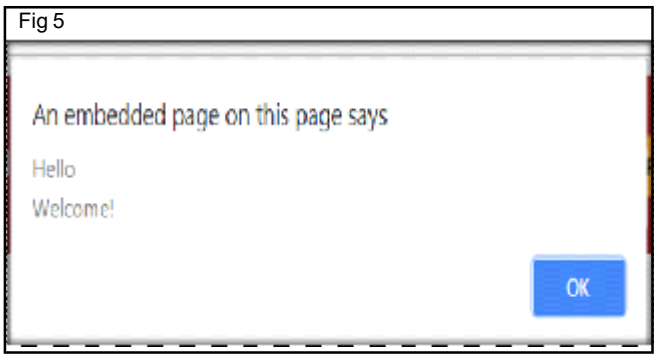
To display line breaks inside a popup box, use a back-slash followed by the character n.

Example 18

alert("Hello\nWelcome!");

The result is shown in Fig 5.

Fig 5



Error handling in JavaScript

Objectives : At the end of this lesson you shall be able to
- **know types of errors in JavaScript**
- **explain error handling in JavaScript.**

Types of Errors

There are three types of errors in programming:

a Syntax Errors

b Runtime Errors

c Logical Errors.

Syntax Errors

Syntax errors, also called **parsing errors,** occur at compile time in traditional programming languages and at interpret time in JavaScript.

For example, the following line causes a syntax error because it is missing a closing parenthesis.

```
<script type="text/javascript">
window.print("Good Morning");
</script>
```

Note : When a syntax error occurs in JavaScript, only the code contained within the same thread as the syntax error is affected and the rest of the code in other threads gets executed assuming nothing in them depends on the code containing the error.

Runtime Errors

Runtime errors, also called **exceptions,** occur during execution (after compilation/interpretation).

For example, the following line causes a runtime error because here the syntax is correct, but at runtime, it is trying to call a method that does not exist.

```
<script type="text/javascript">
window.printme();
</script>
```

Logical Errors

Logic errors can be the most difficult type of errors to track down. These errors are not the result of a syntax or runtime error. Instead, they occur when you make a mistake in the logic that drives your script and you do not get the result you expected.

You cannot catch those errors, because it depends on your requirement what type of logic you want to put in your program.

JavaScript Errors - Throw and Try to Catch

The **try** statement is used to test a block of code for errors. The **catch** statement is used to handle the error. The **throw** statement is used to create custom errors. The **finally** statement is used to execute code, after try and catch, regardless of the result.

When executing JavaScript code, different errors can occur. Errors can be coding errors made by the programmer, errors due to wrong input, and other unforeseeable things.

Example 1

In this example we have written alert as **addalerte** to deliberately produce an error.

```
<p id = "sample"></p>
<script>
try {
addalerte ("Success!");
}
catch (err) {
document.getElement ById("sample").innerHTML = err.message;
}
</script>
```

JavaScript catches **addalerte** as an error and executes the catch code to handle it.

JavaScript try and catch

The **try** statement is used to define a block of code to be tested for errors while it is being executed. The **catch** statement is used to define a block of code to be executed, if an error occurs in the try block.

The JavaScript statements **try** and **catch** come in pairs.

```
try {
    Block of code to try
}
```

```
catch (err){
    Block of code to handle errors
}
```

JavaScript Throws Errors

When an error occurs, JavaScript will normally stop and generate an error message. This is technically called **throw an exception (throw an error)**. JavaScript will actually create an **Error object** with two properties - **name** and **message**.

The throw Statement

The **throw** statement allows you to create a custom error. The exception can be a JavaScript String, a Number, a Boolean or an Object:

```
throw “very small”;    // throw a text
throw 1000;            // throw a number
```

Note: If you use throw together with try and catch, you can control program flow and generate custom error messages.

Input Validation Example

This example examines input. If the value is wrong, an exception is thrown. The exception (err) is caught by the catch statement and a custom error message is displayed:

```
<IDOCTYPE html>
<html>
<body>
<p>please input a number between 40 and 100</p>
<input id = “demo” type = “text”>
<button type = “button”onclick = “myFunction()” >
Sample Input </button>
<p id = “s1”></p>
<script>
function myFunction() {
    var message, x;
    message = document.get Element By Id (“s1”)
    message.inner HTML = “”,
    x = document.getElementById(“demo”).value;
    try {
    if (x ==””) throw “nil”;
    if (isNaN(x)) throw “not a number”;
    x = Number(x);
    if (x < 40) throw “too low”;
    if (x > 100) throw “too high”;
    }
catch (err) {
    message.innerHTML = “Input is” + err;
    }
}
    </script>
    </body>
    </html>
```

Statement

The **finally** statement lets you execute code, after try and catch, regardless of the result:

```
try {
    Block of code to try
}
catch (err) {
    Block of code to handle errors
}
finally {
    Block of code to be executed regardless of the try/
    catch result
    }
```

Example 2

```
function myFunction() {
    var message, x;
    message = document.getElementById(“s1”);
    message.innerHTML = “”;
    x = document.getElementById(“demo”).value;
    try {
    if (x ==””) throw “is empty”;
    if (isNaN(x)) throw “is not a number”;
    x = Number(x);
    if (x>10) throw “is too high”;
    if (x<5) throw “is too low”;
}
catch (err) {
    message.innerHTML = “Error:” + err + “.”;
}
finally {
    document.getElementById (“demo”).value = “”;
    }
}
```

The Error Object

JavaScript has a built in error object that provides error information when an error occurs. The error object provides two useful properties - name and message.

Property		Description
Name	-	Sets or returns an error name
message	-	Sets or returns an error message (a string)

Error Name Values

Six different values can be returned by the error name property. They are

Error Name		Description
Eval Error	-	An error has occurred in the eval() function
Range Error	-	A number "out of range" has occurred
Reference Error	-	An illegal reference has occurred
Syntax Error	-	A syntax error has occurred
Type Error	-	A type error has occurred
URI Error	-	An error in encodeURI() has occurred

Eval Error

An **Eval Error** indicates an error in the eval() function.

Range Error

A **RangeError** is thrown if you use a number that is outside the range of legal values.

Example 3

```
var num = 1;
try {
   num.to Precision (200);   // A number cannot have 200
                                significant digits
}
catch (err) {
   document.get Element By Id ("demo").inner HTML =
   err.name;
}
```

Reference Error

A **ReferenceError** is thrown if you use a variable that has not been declared.

Example 4

```
var x;
try {
    x = y + 1;      // cannot be referenced.
}
catch (err) {
   document.getElementById("demo").innerHTML =
   err.name;
}
```

Syntax Error

A **SyntaxError** is thrown if you try to evaluate code with a syntax error.

Example 5

```
try {

    eval ("alert ("Welcome');   // Missing " will produce
                                   an error
}
catch (err) {
    document.getElementById("demo").innerHTML
     = err.name;
}
```

Type Error

A **TypeError** is thrown if you use a value that is outside the range of expected types.

Example 6

```
var num = 10;
try {
    num.toUpperCase();  // You cannot convert a
                           number to  upper case
        }
catch (err) {
    document.getElementById("demo").innerHTML =
    err.name;
}
```

URI (Uniform Resource Identifier) Error

A **URIError** is thrown if you use illegal characters in a URI function:

Example

```
try {
    decode URI ("%%");     // You cannot use URI
                              decode percent signs
}
catch (err) {
    document.getElementById("demo").innerHTML =
    err.name
}
```

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 element with id="demo":

Example 3

<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:

Array

var person = ["Raja", "Sen", 46];

Objects use names to access its "members". In this example, person.firstName returns Raja:

Object

var person = {firstName:"Raja", lastName:"Sen", age:46};

The 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 type of 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();
```

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
```

Introduction to Function in JavaScript

Objectives : At the end of this lesson you shall be able to
- **define function**
- **explain working of function**
- **explain the benifit 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
}
```

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.

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";
}
```

Objects in JavaScript

Objectives: At the end of this lesson you shall be able to
- **define object**
- **explain object and OOP concept**
- **explain terminology related to objects.**

Introduction to Object-Oriented JavaScript

JavaScript is object-oriented to its core, with powerful, flexible OOP capabilities.

Object-oriented programming

Object-oriented programming is a programming paradigm that uses abstraction to create models based on the real world. It uses several techniques from previously established paradigms, including modularity, polymorphism and encapsulation. Today, many popular programming languages (such as Java, JavaScript, C#, C++, Python, PHP, Ruby and Objective-C) support object-oriented programming (OOP).

Object-oriented programming may be seen as the design of software using a collection of cooperating objects, as opposed to a traditional view in which a program may be seen as a collection of functions or simply as a list of instructions to the computer. In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects. Each object can be viewed as an independent little machine with a distinct role or responsibility.

Object-oriented programming is intended to promote greater flexibility and maintainability in programming and is widely popular in large-scale software engineering. By virtue of its strong emphasis on modularity, object oriented code is intended to be simpler to develop and easier to understand later on, lending itself to more direct analysis, coding, and understanding of complex situations and procedures than less modular programming methods1.

Terminology

Namespace

A container which allows developers to bundle all functionality under a unique, application-specific name.

Class

Defines the characteristics of the object. It is a template definition of variables and methods of an object.

Object

An Instance of a class.

Property

An object characteristic, such as color.

Method

An object capability, such as walk. It is a subroutine or function associated with a class.

Constructor

A method called at the moment of instantiation of an object. It usually has the same name as that of the class containing it.

Inheritance

A class can inherit characteristics from another class.

Encapsulation

A method of bundling the data and methods that use them together.

Abstraction

The conjunction of complex inheritance, methods, properties of an object must be able to simulate a reality model.

Polymorphism

Poly means "many" and morphism means "forms". Different classes might define the same method or property.

For a more extensive description of object-oriented programming, see Object-oriented programming at Wikipedia.

Prototype-based programming

Prototype-based programming is a style of object-oriented programming in which classes are not present, and behavior reuse (known as inheritance in class-based languages) is accomplished through a process of decorating existing objects which serve as prototypes.

This model is also known as class-less, prototype-oriented, or instance-based programming.

The original (and most canonical) example of a prototype-based language is the programming language Self developed by David Ungar and Randall Smith. However, the class-less programming style has recently grown increasingly popular, and has been adopted for programming languages such as JavaScript, Cecil, NewtonScript, Io, MOO, REBOL, Kevo, Squeak (when using the Viewer framework to manipulate Morphic components), and several others.

JavaScript Object Oriented Programming

Namespace

A namespace is a container which allows developers to bundle up all functionality under a unique, application-specific name. In JavaScript a namespace is just another object containing methods, properties and objects.

It is very important to note that there is no language-level difference between regular object and namespaces as there is in some other object-oriented languages.

The idea behind creating a namespace in JavaScript is simple: one global object is created and all variables, methods and functions become properties of that object. Use of namespaces also minimizes the possibility of name conflicts in an application.

Let's create a global object called MYAPP

```
// global namespace

var MYAPP = MYAPP || {};
```

Here in above code sample we have first checked whether MYAPP is already defined (either in same file or in another file). If yes, then use the existing MYAPP global object, otherwise create an empty object called MYAPP which will encapsulate method, functions, variables and objects.

We can also create sub-namespaces:

```
// sub namespace
MYAPP.event = {};
```

Below is code syntax for creating namespace and adding variable, function and method:

```
// Create container called MYAPP.commonMethod for common method and properties

MYAPP.commonMethod = {
regExForName: "", // define regex for name validation
regExForPhone: "", // define regex for phone no validation
validateName: function(name){
// Do something with name, you can access regExForName variable
// using "this.regExForName"
  },
validatePhoneNo: function(phoneNo){
// do something with phone number
}
}
// Object together with the method declarations
MYAPP.event = {
addListener: function(el, type, fn) {
//  code stuff
},
removeListener: function(el, type, fn) {
// code stuff
},
getEvent: function(e) {
// code stuff
 }
// Can add another method and properties
}
//Syntax for Using addListner method:

MYAPP.event.addListener("yourel", "type", callback);
```

Core Objects

JavaScript has several objects included in its core, for example, there are objects like Math, Object, Array and String. The example below shows how to use the Math object to get a random number by using its random() method.

```
alert(Math.random());
```

This and all further examples presume a function named alert (such as the one included in web browsers) is defined globally. The alert function is not actually a part of JavaScript itself.

See JavaScript Reference: Global Objects for a list of the core objects in JavaScript.

Every object in JavaScript is an instance of the object Object and therefore inherits all its properties and methods.

Custom Objects

The Class

JavaScript is a prototype-based language which contains no class statement, such as is found in C++ or Java. This is sometimes confusing for programmers accustomed to languages with a class statement. Instead, JavaScript uses functions as classes. Defining a class is as easy as defining a function. In the example below we define a new class called Person.

```
function Person() { }
```

or

```
var Person = function(){ }
```

The Object (Class Instance)

To create a new instance of an object obj we use the statement new obj, assigning the result (which is of type obj) to a variable to access it later.

In the example below we define a class named Person and we create two instances (person1 and person2).

```
function Person() { }
var person1 = new Person();
var person2 = new Person();
```

Please also see Object.create for a new and alternative instantiation method.

The Constructor

The constructor is called at the moment of instantiation (the moment when the object instance is created). The constructor is a method of the class. In JavaScript, the function serves as the constructor of the object therefore, there is no need to explicitly define a constructor method. Every action declared in the class gets executed at the time of instantiation.

The constructor is used to set the object's properties or to call methods to prepare the object for use. Adding class methods and their definitions occurs using a different syntax described later in this article.

In the example below, the constructor of the class Person displays an alert when a Person is instantiated.

```
function Person() {
  alert('Person instantiated');
}
var person1 = new Person();
var person2 = new Person();
```

The Property (object attribute)

Properties are variables contained in the class; every instance of the object has those properties. Properties should be set in the prototype property of the class (function) so that inheritance works correctly.

Working with properties from within the class is done using the keyword this, which refers to the current object. Accessing (reading or writing) a property outside of the class is done with the syntax: InstanceName.Property; this is the same syntax used by C++, Java, and a number of other languages. (Inside the class the syntax this.Property is used to get or set the property's value.)

In the example below we define the firstName property for the Person class and we define it at instantiation.

```
function Person(firstName) {
  this.firstName = firstName;
  alert('Person instantiated');
}
var person1 = new Person('Alice');
var person2 = new Person('Bob');
// Show the firstName properties of the objects
alert('person1 is ' + person1.firstName); // alerts "person1 is Alice"
alert('person2 is ' + person2.firstName); // alerts "person2 is Bob"
```

The Methods

Methods follow the same logic as properties; the difference is that they are functions and they are defined as functions. Calling a method is similar to accessing a property, but you add () at the end of the method name, possibly with arguments. To define a method, assign a function to a named property of the class's prototype property; the name that the function is assigned to is the name that the method is called by on the object.

In the example below we define and use the method sayHello() for the Person class.

```
function Person(firstName) {
this.firstName = firstName;
}
Person.prototype.sayHello = function() {
alert("Hello, I'm " + this.firstName);
};
var person1 = new Person("Alice");
var person2 = new Person("Bob");
// call the Person sayHello method.
person1.sayHello(); // alerts "Hello, I'm Alice"
person2.sayHello(); // alerts "Hello, I'm Bob"
```

In JavaScript methods are regular function objects that are bound to an object as a property, which means they can be invoked "out of the context". Consider the following example code:

```
function Person(firstName) {
this.firstName = firstName;
}
Person.prototype.sayHello = function() {
alert("Hello, I'm " + this.firstName);
};
```

```
var person1 = new Person("Alice");
var person2 = new Person("Bob");
var helloFunction = person1.sayHello;
person1.sayHello();  // alerts "Hello, I'm Alice"
person2.sayHello();  // alerts "Hello, I'm Bob"
helloFunction();      // alerts "Hello, I'm undefined" (or fails
                      // with a TypeError in strict mode)
alert(helloFunction === person1.sayHello);        // alerts true
alert(helloFunction === Person.prototype.sayHello); // alerts true
helloFunction.call(person1);  // alerts "Hello, I'm Alice"
```

As that example shows, all of the references we have to the sayHello function - the one on person1, on Person.prototype, in thehelloFunction variable, etc. - refer to the same function. The value of this during a call to the function depends on how we call it. In the common case when we call it in an expression where we got the function from an object property - person1.sayHello() - this is set to the object we got the function from (person1), which is why person1.sayHello() uses the name "Alice" and person2.sayHello() uses the name "Bob". But if we call it other ways, this is set differently: Calling it from a variable - helloFunction() - sets this to the global object (window, on browsers). Since that object (probably) doesn't have a firstName property, we end up with "Hello, I'm undefined". (That's in loose mode code; it would be different [an error] in strict mode, but to avoid confusion we won't go into detail here.) Or we can set this explicitly using Function#call (or Function#apply), as shown at the end of the example.

Inheritance

Inheritance is a way to create a class as a specialized version of one or more classes (JavaScript only supports single inheritance). The specialized class is commonly called the child, and the other class is commonly called the parent. In JavaScript you do this by assigning an instance of the parent class to the child class, and then specializing it. In modern browsers you can also use Object.create to implement inheritance.

JavaScript does not detect the child class prototype.constructor (see Object.prototype), so we must state that manually.

In the example below, we define the class Student as a child class of Person. Then we redefine the sayHello() method and add thesayGoodBye() method.

```
// Define the Person constructor
function Person(firstName) {
  this.firstName = firstName;
}
// Add a couple of methods to Person.prototype
Person.prototype.walk = function(){
  alert("I am walking!");
};
Person.prototype.sayHello = function(){
  alert("Hello, I'm " + this.firstName);
};
// Define the Student constructor
function Student(firstName, subject) {
// Call the parent constructor, making sure (using Function#call) that "this" is
// set correctly during the call
Person.call(this, firstName);
// Initialize our Student-specific properties
this.subject = subject;
};
// Create a Student.prototype object that inherits from Person.prototype.
// Note: A common error here is to use "new Person()" to create the Student.prototype.
// That's incorrect for several reasons, not least that we don't have anything to
// give Person for the "firstName" argument. The correct place to call Person is
// above, where we call it from Student.
Student.prototype = Object.create(Person.prototype); // See note below
// Set the "constructor" property to refer to Student
Student.prototype.constructor = Student;
// Replace the "sayHello" method
Student.prototype.sayHello = function(){
alert("Hello, I'm " + this.firstName + ". I'm studying " + this.subject + ".");
};
// Add a "sayGoodBye" method
Student.prototype.sayGoodBye = function(){
alert("Goodbye!");
};
// Example usage:
var student1 = new Student("Janet", "Applied Physics");
student1.sayHello();   // "Hello, I'm Janet. I'm studying Applied Physics."
student1.walk();       // "I am walking!"
student1.sayGoodBye(); // "Goodbye!"
```

// Check that instanceof works correctly

alert(student1 instanceof Person); // true

alert(student1 instanceof Student); // true

Regarding the Student.prototype = Object.create(Person.prototype); line: On older JavaScript engines without Object.create, one can either use a "polyfill" (aka "shim", see the linked article), or one can use a function that achieves the same result, such as:

function createObject(proto) {

function ctor() { }

ctor.prototype = proto;

return new ctor();

}

// Usage:

Student.prototype = createObject(Person.prototype);

See Object.create for more on what it does, and a shim for older engines.

Encapsulation

In the previous example, Student does not need to know how the Person class's walk() method is implemented, but still can use that method; the Student class doesn't need to explicitly define that method unless we want to change it. This is called encapsulation, by which every class inherits the methods of its parent and only needs to define things it wishes to change.

Abstraction

Abstraction is a mechanism that permits modeling the current part of the working problem. This can be achieved by inheritance (specialization), or composition. JavaScript achieves specialization by inheritance, and composition by letting instances of classes be the values of attributes of other objects.

The JavaScript Function class inherits from the Object class (this demonstrates specialization of the model) and the Function.prototype property is an instance of Object (this demonstrates composition).

var foo = function(){};

alert('foo is a Function: ' + (foo instanceof Function)); // alerts "foo is a Function: true"

alert('foo.prototype is an Object: ' + (foo.prototype instanceof Object)); // alerts "foo.prototype is an Object: true"

Polymorphism

Just like all methods and properties are defined inside the prototype property, different classes can define methods with the same name; methods are scoped to the class in which they're defined. This is only true when the two classes do not hold a parent-child relation (when one does not inherit from the other in a chain of inheritance).

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

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 numhas 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 numhas 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 numhas 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 numhas 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 numhas 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 cone 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

t**oString()** 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.

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 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

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)

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.

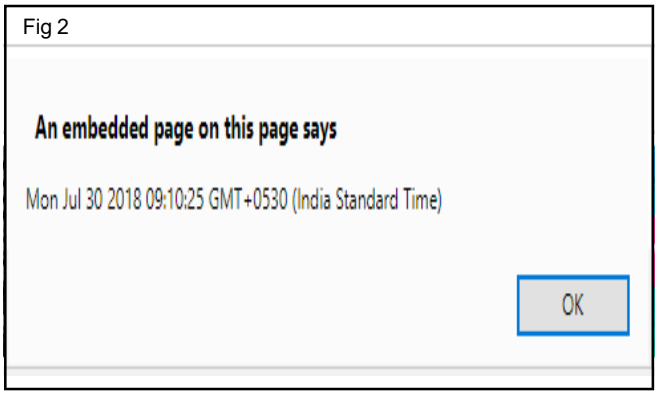
Example 2

```
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)

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);
```

Example 5

```
var d = new Date(2018, 06, 30, 09);
```

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)

Fig 3

You cannot omit month. If you supply only one parameter it will be treated as milliseconds.

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

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)

Fig 5

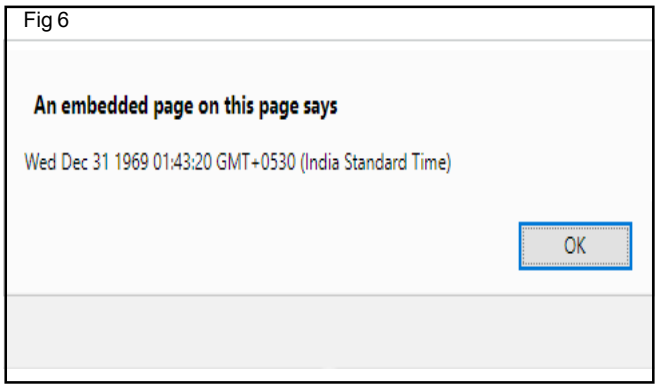
January 01 1970 minus 100 000 000 milliseconds is approximately December 31 1969.

Example 13

var d = new Date(-100000000);

Result: (Fig 6)

Fig 6



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)

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)

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)

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)

Fig 10

The getMonth() Method

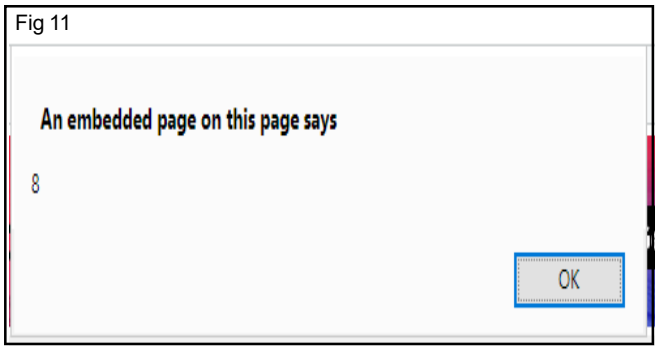
The getMonth() method returns the month of a date as a number (0-11).

Example 30

```
var d = new Date();

document.getElementById("demo").innerHTML =
d.getMonth();
```

Result: (Fig 11)

Fig 11



In JavaScript, the first month (January) is month number 0, so December returns month number 11.

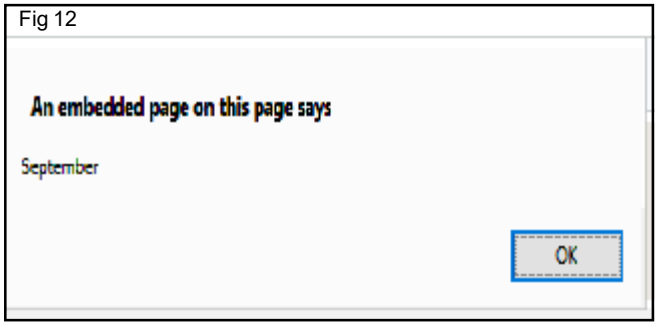
You can use an array of names, and getMonth() to return the month as a name.

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)

Fig 12



The getDate() Method

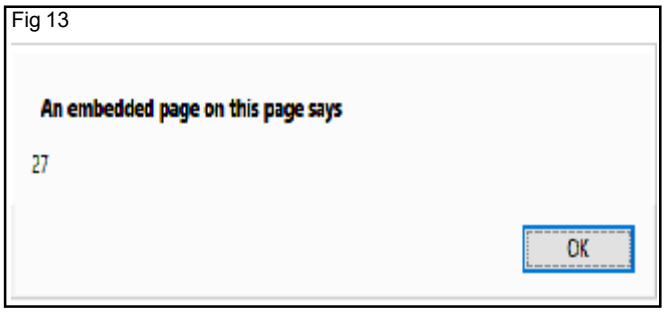
The getDate() method returns the day of a date as a number (1-31).

Example 32

```
var d = new Date();

alert(d.getDate());
```

Result: (Fig 13)

Fig 13



The getHours() Method

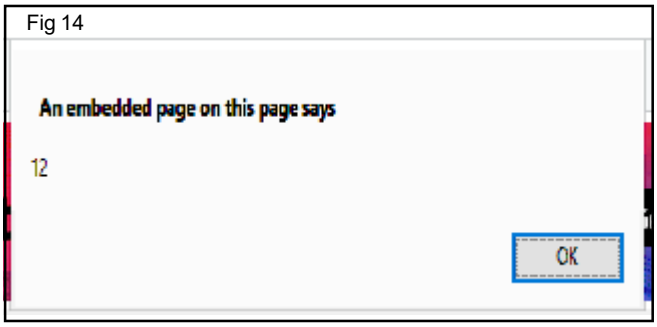
The getHours() method returns the hours of a date as a number (0-23).

Example 33

```
var d = new Date();

alert(d.getHours());
```

Result: (Fig 14)

Fig 14



The getMinutes() Method

The getMinutes() method returns the minutes of a date as a number (0-59).

Example 34

```
var d = new Date();

alert(d.getMinutes());
```

Result: (Fig 15)

Fig 15

The getSeconds() Method

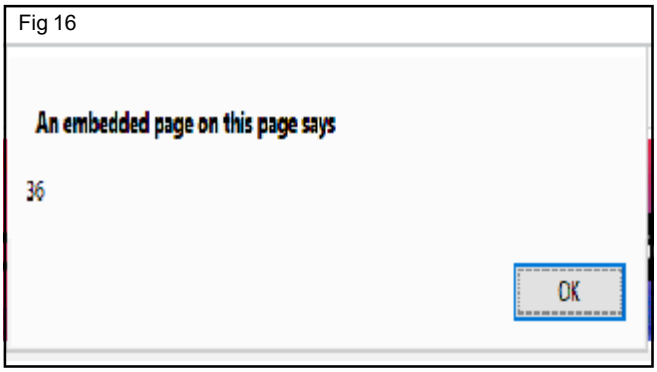
The getSeconds() method returns the seconds of a date as a number (0-59).

Example 35

```
var d = new Date();

alert(d.getSeconds());
```

Result: (Fig 16)

Fig 16



The getMilliseconds() Method

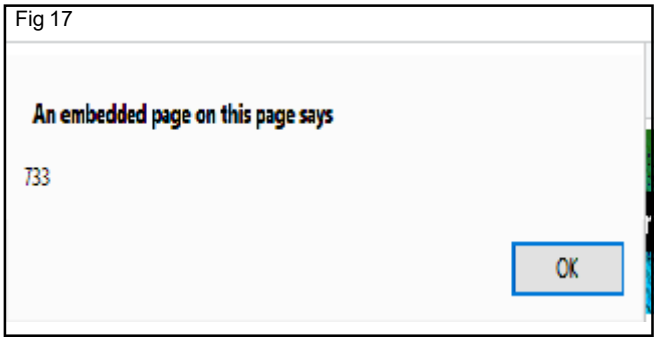
The getMilliseconds() method returns the milliseconds of a date as a number (0-999).

Example 36

```
var d = new Date();

alert(d.getMilliseconds());
```

Result: (Fig 17)

Fig 17



The getDay() Method

The getDay() method returns the weekday of a date as a number (0-6).

Example 37

```
var d = new Date();

alert(d.getDay());
```

Result: (Fig 18)

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"

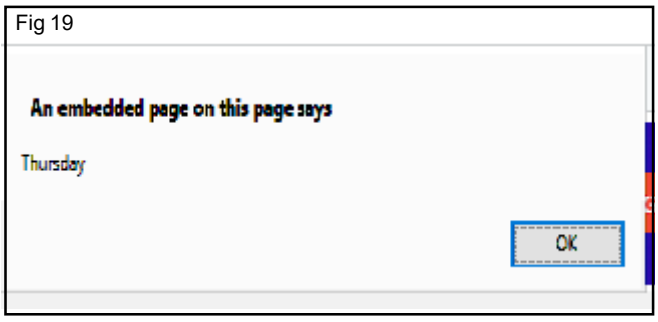
You can use an array of names, and getDay() to return the weekday as a name.

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
getUTCMinutes()	Same as getMinutes(), but re turns the UTC minutes
getUTCMonth()	Same as getMonth(), but returns the UTC month
getUTCSeconds()	Same as getSeconds(), but re turns 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)

Fig 22

The setDate() Method

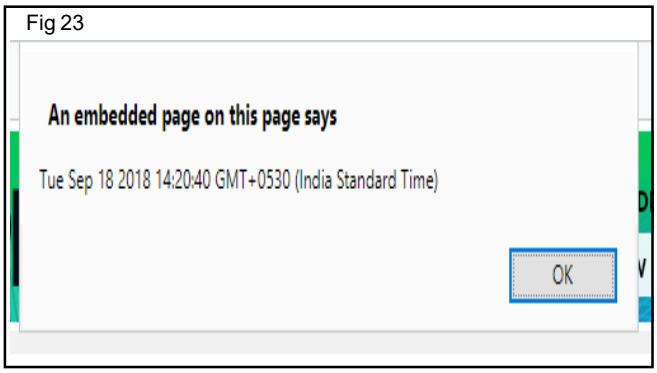
The setDate() method sets the day of a date object (1-31).

Example 42

```
<script>
var d = new Date();
d.setDate(18);
alert(d);
</script>
```

Result: (Fig 23)

Fig 23



The setDate() method can also be used to add days to a date.

Example 43

```
<script>
var d = new Date();
d.setDate(d.getDate() + 25);
alert(d);
</script>
```

Result: (Fig 24)

Fig 24

If adding days, shifts the month or year, the changes are handled automatically by the Date object.

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)

Fig 25

The setMinutes() Method

The setMinutes() method sets the minutes of a date object (0-59).

Example 45

```
<script>
var d = new Date();

d.setMinutes(15);

alert(d);

</script>
```

Result: (Fig 26)

Fig 26

The setSeconds() Method

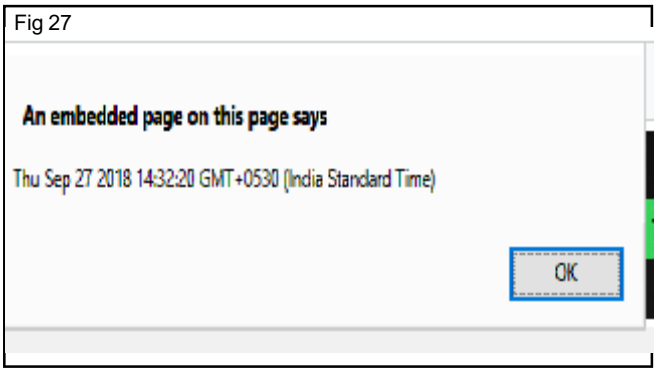
The setSeconds() method sets the seconds of a date object (0-59).

Example 46

```
<script>
var d = new Date();
d.setSeconds(20);
alert(d);
</script>
```

Result: (Fig 27)

Fig 27



Compare Dates

Dates can easily be compared.

The following example compares today's date with January 14, 2100.

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 greather then Date Two.");
 }
 else
 {
 alert("Date Two is greather then Date One.");
  }
```

Document Object Model and Open source software

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.

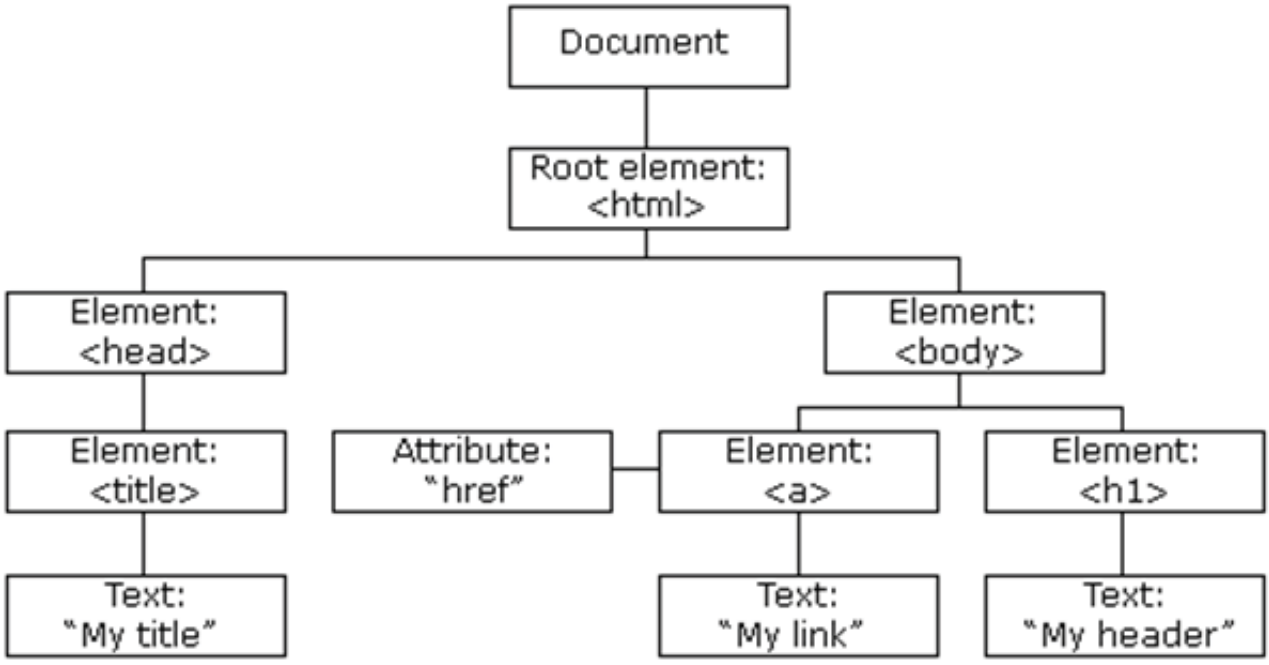
The HTML DOM (Document Object Model)

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!";
</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.

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

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 get ElementsByClassName()

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.documentMode	Returns the mode used by the browser	3
document.documentURI	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 <img> element.

Example 9

```
<!DOCTYPE html>
<html>
<body>
<img id="Image1" src="flower.gif">
<script>
document.getElementById("Image1").src = "newflower.jpg";
</script>
</body>
</html>
```

Example explained:

- The HTML document above contains an <img> 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

- 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/.

Develop and edit web pages in KompoZer

Objectives : At the end of this lesson you shall be able to
- **know introduction to KompoZer**
- **know develop and edit web pages**
- **know the use of templates**
- **explain Publishing web sites**
- **describe Site Manager**
- **know preferences**
- **explain publishing web sites.**

Introduction

What is KompoZer?

KompoZer is a complete Web Authoring System which integrates web page development and web file management. It provides a web page editor which has a simple graphical (WYSIWYG - what you see is what you get) interface. With KompoZer, newcomers will quickly and easily be able to produce new web pages. The output code is compliant to a high extent with the latest issues of the appropriate web language specifications

KompoZer incorporates a Site Manager; this gives rapid access to the files on both local machines and remote servers. It can cater for several sites and switch rapidly between them. From within KompoZer pages and associated files may be uploaded to a remote server. KompoZer supports the use of "Styles" through Cascading Style sheets (CSS) both embedded and external. It has an editor which generates CSS code conforming with CSS 2.1 specifications.

Who is KompoZer for?

KompoZer is suitable for anyone wishing to have a modern, free of charge, program for developing small web sites and who would like to learn modern web design techniques such as the use of CSS.

Basics :

Open KompoZer. The main window opens. At the top are a number of toolbars. The topmost is the Menu Bar. This carries a number of items (File, Editetc) used to make selections. The next is the 'Composition Toolbar' which carries a number of 'Buttons' labelled 'New', 'Open' etc.

To create a new page: On the Composition toolbar Click the 'New' button.

To open an existing page:Assuming that the page is stored on your local disk in HTML format: On the menu Bar click 'File' then 'Open File'. Browse to the file and click 'Open'.

Editing a web page: Your web page - blank or otherwise - is in the large pane in the centre right of the KompoZer application window. Many editing functions are very similar to those in a word processor. The top four toolbars on the KompoZer application window provide a number of editing functions - to see what any do hover the cursor over an item and a hint will appear.

Saving a Page To save a page: On the Composition toolbar click 'Save'. If it was a new document a dialog window will ask you to enter a title for the page. This will appear in the tab at the top of the page display area. NB this is NOT the file name. Click 'OK'; you will then be offered a normal save window which allows you to browse to a suitable location and name the file. The file extension offered will be HTML.

Browsing a page :To see how your page will look in your default browser on the Composition toolbar click 'Browse'.

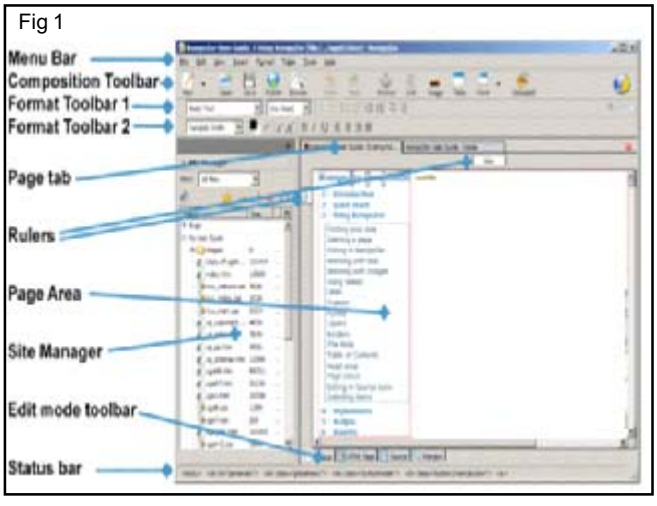
Help: The help system should be a first resort in case of need. The forum at http://wysifauthoring.informe.com/ is a place for sharing experiences and obtaining and giving help. Many of the contributors are users of the programs and range from beginners to those with lots of experience.

Develop and edit Web pages

KompoZer Screen Layout

KompoZer screen layout. (Fig 1)

Fig 1



When KompoZer starts the window carries a menu bar across the top (File, Edit, View etc). Below this are three 'Toolbars'. To ensure that everything is visible, on the Menu bar select**View > Show/Hide** and see that each of the following is checked:

- Composition Toolbar,
- Format Toolbar 1
- Format Toolbar 2
- Edit Mode Toolbar
- Status bar
- Rulers
- Site Manager

The three toolbars across the top of the window carry buttons (represented by icons). Hover on any to find its function. If any are greyed out they are not functional in the current mode as they are context sensitive. Across the centre of the window are two panes: the 'Site Manager' on the left and a blank web page on the right. The Site Manager is a powerful tool. Since it is not needed yet it may be closed by clicking on its close button or pressing F9. At the top of the Page area there is a Page tab which carries the name of the page ('Untitled'). KompoZer Help refers to this as the 'Tab Browser' toolbar. If you had several pages open, as shown in the Fig.1, this tab would allow you to select one them rapidly. At the bottom of the page area is the 'Edit Mode Toolbar' which carries four tabs which select one of four 'Viewing modes' for a page ('Normal', 'HTML Tags', 'Source', 'Preview'.)

At the bottom of the window is the 'Status Bar'. This is a very powerful tool. Once a page is populated, by clicking any item in the page area its structure appears on the status bar. Any class or id applied to an element is shown and any bearing an inline style isindicated in italic type. Hovering reveals the style declaration. Additionally clicking an element marker highlights the element in both normal view and, on changing view, in source view thus simplifying navigation in source view. Note The figure shows the buttons as they appear when KompoZer is first installed. They may be customised to different arrangements. If this has been done some of the following may be difficult to follow.

To restore defaults click **View > Show/Hide > Customize Toolbar > Main Toolbar > Restore Default Set** and repeat similarly for the Format Toolbar. Using KompoZer right-click any toolbar to customise it.

Options for starting a page

There are several ways to start new pages or open existing ones.

To start a new blank page, on the menu bar click File > New. A window headed 'Create a new document or template' appears. Check the boxes 'A blank document' and 'Strict DTD' and clear 'create a XHTML document'.

To open an existing page

1 Click the OPEN button to access a normal browse dialogue.

2 Click File > Recent Pages to get rapid access to those recently worked on.

3 Or use the Site Manager which provides a powerful mini-browser and is very easily set up. The Doctype of an existing page will remain the same as before it was opened. It cannot be changed in KompoZer.

Each page opened starts in a new tab which can be clicked to select a document to work on.

Editing in KompoZer

KompoZer supports all the standard Windows editing commands and shortcut keys. e.g. Copy Ctrl+C, Paste Ctrl+V etc. There are other KompoZer specifics. These are great time savers. In 'HTML Tags' view KompoZer supports drag and drop editing for block items. (Select an item by pressing the Control key while clicking on the Tag.) An extreme time saver is KompoZer's double click response. In several cases, such as links, images and tables, a very useful editing window is opened. KompoZer supports many levels of Undo and Redo, however changes made in 'Source' view cannot be undone after you have changed the view.

Saving files

Go to File > Save as. You are offered a 'Save Page as' window which allows you to browse to the folder you want to use. You will find the file name already completed with your page title. You will probably want to changethis to a shorter, all lower case, name. You will find the file extension completed as 'html' you may prefer to, and may alter it to "htm".

Printing pages

The 'Print' button allows you to print the current page to a printer. This prints the page view and not the source code.

Working with text

Text typed directly onto the KompoZer page defaults to appearing in the format for the 'Body' element.

HTML defines a small number of elements specifically for text and it is usually preferable to use these. To format text in a standard element format select the text and click the first drop-down box on the format toolbar. This offers a selection of standard text formats. Paragraph is the most appropriate for general text.

Once formatted as a paragraph, when typing in a text area, use of the 'enter' key starts a new block of text i.e. a paragraph. To start a new line within the current paragraph press Shift+Enter; this generates a line break.

Other standard text formats are Heading formats from Heading 1, the largest (for the main heading), to Heading 6, the smallest (for the least significant). Browsers generally render headings in bold type. Text can be edited in any of the viewing modes and KompoZer responds to all the normal windows shortcut commands.

Formatting text

Text can be formatted in a number of ways using a format toolbar. The changes listed in the table can be applied (hover over a tool to discover what it does).

Choose a font , Choose text colour, Choose background colour, Choose highlight colour, Make text smaller, Make text larger, Embolden, Italicise, Underline, Format as a numbered list, Format as a bulleted list, Align left, Align Centre, Align right, Justify, Indent text, Outdent text,Emphasise, Strongly emphasise.

Numbered and Bulleted lists

KompoZer can format a list of items giving each item a sequential number in any of several formats or presenting them bulleted. To start a list from scratch

1 Click one of the list buttons (Numbered List or Bulleted list) on the Format toolbar.

2 Type the first item.

3 Press Enter and type the next item.

4 To finish, on the last(blank) item press Enter.

To change existing text into a list

1 Select the text required.

2 Click one of the list buttons on the Format Toolbar. The text will be changed into a list, a new item starting for each paragraph or other block item encountered.

To add items to a list

1 Click at the end of the last item in the list.

2 Press Enter and type the new item. Numbering and format will continue from the previous item.

Importing text

Strictly KompoZer does not support importing text from other applications but it is possible to copy and paste text. In normal view content from other web pages may be copied reliably. The result will be rendered according to any styling applied in your document; any reliance on external styles in the original document will be lost.

Text from word processors such as Microsoft Word or OpenOffice.org in rtf or doc format or from text editors such as Windows notepad may also be copied and pasted. When such text is pasted into KompoZer most formatting is lost. Numbered lists will be retained.

Tabs will be rendered as three non-breaking spaces. The contents of tables may be pasted, individual cells will be separated as if by tabs.

Special characters

By special characters we refer to characters which do not have an equivalent keyboard key. HTML uses a system of characters known as 'Unicode'. This includes a large range of characters including all the international currency symbols and many thousands of others, though the fonts supplied on computers will support only a subset. A number of the more common, including accented ones, may be inserted using Insert > Characters and Symbols.

Checking spelling

In any view, other than Source, click on the 'Spell' button.

 .The spell checker will work sequentially through the page.

To insert an image

1 Click the 'Image' button on the Composition toolbar

2 The Image properties window opens. Click 'Choose File' and browse and select a file

3 Click 'Open'. Leave checked the box 'URL is relative to page location' this will allow you to move the page and image to another location, as you will have to when you upload them to a server. (If you de-select this and move the page, it will try to find the image at the original location.) Note If the box is 'greyed out' this is probably because the page has not been saved.

4 In the box labelled 'Alternate text' add a description of the image. (This forms the 'alt' attribute for the image and provides text which will appear in place of the image with user agents (browsers) that cannot display images (screen readers and voice synthesisers). It will also be used by those with visual impairment. The content of this box must be carefully considered so as to be of maximum assistance in such cases.) Note Where the image is purely decorative, and not necessary to understanding the page, alternative text is not required and should be omitted

5 In the box labelled 'Tooltip' you may optionally insert a 'Title' attribute for the image. Some browsers will show the text provided when the cursor hovers over the image

6 Click OK

Using tables

Tables allow data - images, links, forms, form fields, text, etc. - to be arranged into rows and columns of cells. Inserting tables

1 On the Composition Toolbar click the Table button. The 'Insert table' window appears

2 Leave the 'Quickly' tab selected and drag out a matrix then click the bottom right cell to define the table arrangement

3 The cells appear on the screen with narrow outlines

Note: If later the table border is set to zero these outlines disappear but KompoZer in normal view replaces them with a red outline. This does not appear in Preview or in a browser. Tables have resizing boxes similar to those used with images

Table cell properties

Right-click the table and select 'Table Cell properties'. The Table properties window opens. This has two tabs 'Table' and 'Cells' which allow overall control of several aspects of either the table or individual cells. This includes

a Alignment of text within cells

b Wrapping of text

c Cell spacing - the gap between cells

d Cell padding - the gap between the edge of the cell and the text within it

e Size of table and cells

f Background colour

g Selection of cells as 'Normal' or 'Header' (Cells which are headings to rows or columns should be selected as 'Header'. Normally this results in them being rendered in bold type.)

Linking text

Linking to another file

To create a link

1 Select (highlight) a few words of text

2 On the Composition toolbar click on the 'Link' button, alternatively Right-click and select 'Create Link'. The 'Link Properties' window opens

3 Click on 'Choose File' and browse to the file that you want to link to

4 Click OPEN

5 Click OK

Inserting an email address

Instead of linking to a file it is possible to insert an email address. The result will be that, in use, when the link is clicked the email client on the visitor's machine will be opened with the correct address selected.

To do so proceed as under the previous heading. When the Link Properties window opens (or if Image Properties click the Link tab) enter the email address and check the box 'The above is an email address'.

Inserting named Anchors

There is a second type of Anchor element the 'Named anchor'. Such an anchor is extremely useful as it can act as a type of bookmark defining a particular place on a page. Links can jump to such bookmarks.

To insert a named anchor

1 Place the cursor at the point you want to mark.

2 Click the 'Anchor' button on the Composition toolbar or, on the Menu Bar, select Insert > Named Anchor. The named anchor properties window appears.

3 Enter a unique name for the anchor.

4 Click OK. In 'Normal' view anchors are marked by a picture of an anchor  .

Linking to named anchors

Start as above for linking to another file. When the 'Link Properties' window opens, instead of choosing file use the drop down list. Your anchor name should appear there preceded by a "#". Click it and OK. That's it! If you test your page on a browser when you click the link the view should move to show the position of the anchor.

Linking images

The techniques and possibilities are very similar to those used with text.

To create a link

1 Click on the image

2 On the Composition toolbar click on the 'Link' button, alternatively Right-click and select 'Create Link'. The 'Image Properties' window opens

3 Click on 'Choose File' and browse to the file that you want to link to. (The box 'URL is relative to page location' is checked. This means that if you move your page to a new folder you should move the image to a corresponding new folder. If you clear this box the absolute address of the image on the hard drive is given in full. If you move your page now it will look there for the image. As you start to organise a web site you will find that this is not a good arrangement and potentially disastrous when you upload the page to a server.)

4 Click OPEN

5 Click OK

Editing Links

To change the file to which a link refers, in Normal, Tags or Preview mode double-click on the link. The 'Link properties' window opens (for an image the 'Image Properties' window opens - click the Link tab). Edit the link.

To remove the link delete the link reference in the box.

Frames

If you open a frame document, you get a message 'This page can't be edited for an unknown reason' but it displays the frame content rather beautifully. Then you can do nothing with it except click on the 'Source' tab. You then see the source code and the system will seem to lock up. Actually it doesn't lock and you can load another page and revert to normal operation.

This is not a great limitation. Though the code for frames takes a little getting used to it is usually quite short and can easily be produced using a text editor. Once established it probably rarely needs to be altered. You can use KompoZer to develop the pages that go into the frames.

Forms

To set up a form

1 Click the form button.

2 In the Form properties window give the form a name of your choosing

3 Complete the Action box with the correct URL and select the appropriate method

4 'Encoding' and 'Target Frame' will frequently not be required but, if they are, select 'More Properties' and complete the boxes

5 Click OK

6 On the form place any headings, paragraphs and images ensuring that there is a placeholder for any controls needed. (If blank placeholders are needed it is probably sensible to put some dummy text in now and delete it later.)

7 Where controls are needed click the corresponding placeholder and using the drop down box beside the Form button select the required control

8 Give each control a unique name

9 Each control has specific information which needs to be entered. Enter it into the box in the window which appears

Table of contents

If you have a long document with sections headed using heading formats Heading 1, Heading 2 etcKompoZer can generate a Table of Contents (ToC) automatically. The table reflects the structure of the page, the content of the headings forming the text of the table.

Inserting a Table of Contents

1 Place the cursor where the table is required

2 Click Insert > Table of Contents > Insert

3 The 'Table of Contents' window appears

4 In the column headed 'Tag' select the tag for each level e.g. against level 1 select h3 and against level 2 select h4 and for all the others select '--'

5 If, instead of using headings you wish to use classed paragraphs or a div, instead of selecting a heading tag select 'p' or 'div' and in the box in the column headed 'Class' enter the class required. (It is, of course, also possible to select headings by allocating a class.)

6 If you wish the contents to be numbered check the box 'Number all entries ...'

7 Click OK

The Table of Contents will be created.

To update a ToC

After changes have been made to a page. There is no need to place the cursor.

1 Click Insert > Table of Contents > Update

2 The 'Table of Contents' window appears showing the selections previously made

3 If desired, changes may be made to the selections

4 To update the ToC click OK

To delete a ToC

There is no need to place the cursor.

1 Click Insert > Table of Contents > Remove

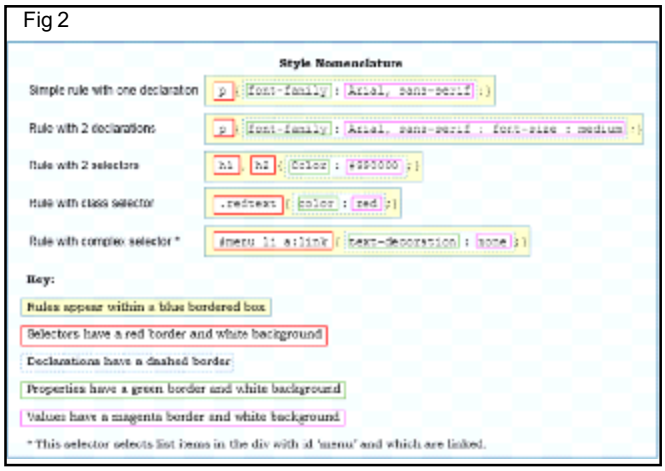
Style and stylesheets

Styles specify how particular elements on a page appear on the screen, in print or whatever. Style may typically define aspects of presentation as the font face, size and variant, the font colour, the background colour, whether an element is to be aligned right, centre or left, whether spaced away from other elements, surrounded by a border and, if so, what type or colour. Elements may be given an absolute position relative to the page (which allows elements to overlap).

Elements such as paragraphs, tables and images are considered to exist within boxes or blocks and the sizes of these boxes may be specified.

Classes As well as allowing you to specify the style of elements it is possible to define styles and apply them selectively to only some elements. This is done through 'Classes' - a 'Class' is just a style that can be applied as and when you choose. 'Classes' are applied to 'Tags' (a marker attached to an element). The element to which the class is applied appears in the format defined by the class. Other similar elements without the class applied appear in the default format i.e. either the default specified by the browser or the style that the user has defined for the corresponding element. On the status bar KompoZer shows classes along with the tag to which they are applied. Fig 2 shows Style Nomenclature.

Fig 2



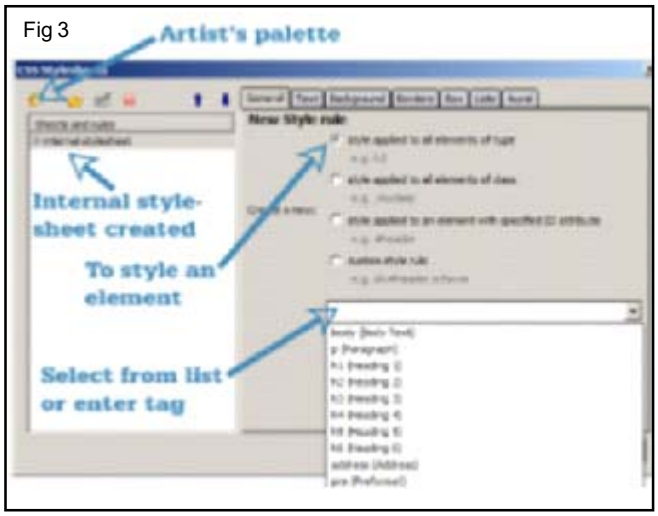
Creating styles

Internal styles: AsKompoZer includes a CSS editor called CaScadeS which allows you to create stylesheets and define rules. Before you define a rule you must have a stylesheet but if you are working on a page which has none CaScadeS will create one for you.

Creating a style rule for an element.

Css Stylesheet window (Fig 3)

Fig 3



To create a style rule for an element

1 Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens.

2 Click on the artist's palette button. In the 'Sheets and rules' pane you will see an internal stylesheet has been created for you.

3 To create a rule click 'Style applied to all elements of type'

4 Beside the blank box click the drop down arrow. You will see listed a number of common elements. To create a style for one of these click it alternatively enter the tag for any other element.

5 Click 'Create Style rule'

6 You are now presented with a window headed'Selector' followed by the tag for the element. The window actually lists the style declarations for that element, but of course that is now blank.

To define how you want elements to look

1 Select in turn as required the tabs for 'Text', 'Background', 'Border' etc and specify exactly how you wish that element to appear. The next section amplifies some details of how to do this.

2 Return to the general tab to see the full declarations that you have set for the Selector.

3 If you click the 'General' tab you will see all the declarations for the rule. You can edit these here but it is better to leave the job to CaScadeS because if you make any errors the declaration will be deleted

4 When you are satisfied click 'OK'.

Link to an existing stylesheet

If you have a stylesheet that you created for another page or intend to use right across your site you can link your page to that.

To link to an existing external style sheet

1 Click the CaScadeS button on the Composition toolbar. The CSS Stylesheets window opens

2 Click the dropdown arrow beside the artist's palette button and select 'Linked stylesheet'

3 Click 'Choose file' and browse to the file that you need

4 Click 'Open'

5 Click 'Create stylesheet'

6 Click 'OK' You can now close CaScadeS, but, of course you can work on the stylesheet in the usual way.

Saving stylesheets

Once you edit a stylesheet in KompoZer its name is marked by a red symbol indicating that it has not been saved. When you close CaScadeS changed sheets are saved immediately.

Removing styles

CaScadeS allows you to remove styles in a similar way to adding them. In the 'Sheets and rules' pane select the rule you want to remove and click 'Remove'. Similarly you can remove a stylesheet. Select the sheet and click 'Remove'. If you select an internal stylesheet it is deleted from the file completely.

Templates

Templates are basically pages having some content (e.g. a letter head) which can be re-used to create other pages which will have the same underlying page structure and, often, the same graphical layout.

Templates are not altered in use and can be used over and over again. The simplest template is probably a blank sheet which links to a stylesheet for use throughout a site. More common is a page which has a banner and perhaps a menu to appear on every page. Last might be a complete page layout for use on all, or many, pages of a site but which includes areas for customising individually. Templates may be considered as having two parts - the fixed part or 'boilerplate' which remains the same for every page and the 'editable part' which changes.

Create a template from a page

A pre-existing document may be transformed into a template

1 Click Format > Page Title and Properties

2 Check the box 'This page is a template'.

3 Click OK.

4 Click File > Save as. The file type 'HTML Template' will be completed.

5 Name and save the file as normal

To save a template

1 Click File > Save or File > Save As.

2 The extension 'mzt' will be selected automatically.

To make blocks editable

1 In turn, select each block that you wish to make editable.

2 In HTML tags view select the block by clicking its tag.

3 On the status bar right-click the corresponding highlighted tag.

4 Click Templates > Make editable.

5 In the 'Insert an editable area' window give the block a recognisable name. Now check the options boxes if required.

6 Click OK.

To make a flow selection editable

1 In turn select (highlight) each section of text that you wish to make editable.

2 Click Insert > Templates >Insert editable area.

3 In the 'Insert an editable area' window give the block a recognisable name.

4 Leave checked the option 'Flow of text'.

5 Check the options boxes if required as described above. Note The option 'Area is moveable' is inappropriate for flow areas).

6 Click OK

To create a page using Template

1 Click File > New > A new document based on a template > Choose File.

2 Select the Template (Note templates have the file extension 'mzt')

3 Click 'Create'.

To use the page

1 Click in turn in each editable areas.

2 Select and delete the sample text and replace it with new text.

2a If the editable area was repeatable a small square appears within the label, hovering turns it red and clicking makes a copy. Copies have small circles which act as delete buttons.

2b If any area was optional a small circle with an x appears within the label. Hovering turns it red and clicking deletes it. The same figure shows this for the 'Other languages'. Because of the limitation described it has not been possible to fill in all the editable areas.

3 When all editable boxes have been completed detach the page from the template by clicking Edit > Detach from template. The page now assumes its final appearance.

4 Save the page in the normal way. The last figure shows the final result. The areas which could not be edited earlier have been completed. Now it is possible to edit any item and as a workaround the frozen repeatable items may be added.

Editing templates

Templates which have already been saved may be altered after opening using menu commands File > Open File and selecting 'Files of Type' then 'HTML Templates'.

Site Manager

The site manager allows you to navigate your site or between sites easily. To toggle the Site Manager on or off either press F9 or use View > Show/Hide > Site Manager. Site Manager can deal with sites irrespective of whether they reside on a local machine or on a remote server. In the latter case, if you are on a dial-up network, Site Manager will dial and make the connection for you. Since generally you will set up a site on a local machine and later 'publish' to a remote server we will deal first with setting up on a local machine. Site Manager provides a directory tree view of a site similar to the view with Windows Explorer. It however lists only directories which you have specifically set up as 'Sites'. You can set up many sites, they appear in Site Manager irrespective of where they appear in a normal directory tree.

Setting Preferences

You can set up a number of features in KompoZer according to personal preferences. Several of the options are grouped under the Tools >Optons menu. In addition you can customise toolbars via the View > Show/hide menu.

Defaults

The defaults set by KompoZer will generally be found satisfactory. (Fig 4)

Fig 4

Menu selection	Default
Tools > Options > General	
Maximum number of pages	10
Retain original source formatting	Checked
Reformat HTML Source	Cleared
Save images when saving pages	Cleared
Always show publish dialog	Cleared
Maintain table layout	Checked
Use CSS styles	Checked
Always open a document in a new tab	Checked
Tools > Options > Fonts	
Allow documents to use other fonts for others see text	Checked
Tools > Options > New Page settings	
Author	Blank (see below)
Reader's default colors	Checked
Background image	Blank (see below)
Language	Blank (see below)
Writing direction	No direction specified
Character set	ISO-8859-1
Tools > Options > Advanced	
Set up Proxies	Direct connection
Markup - Language	HTML 4
Markup - DTD	Strict
Return in paragraph always creates new	Checked
Underline misspelled words	Checked
Output the following characters	HTML4
Special characters	Only & < > and no break space
Don't encode '>' outside attribute	Cleared
Don't encode special characters	Cleared

Editing preferencesKompoZer can be customised in several ways through the Menu selection Tools > Options mechanism. All the options may be set at any time. All take effect immediately except for 'New page settings' which do not apply to any existing page.

Publishing

Introduction Publishing a site means transferring the site, i.e. the pages, images and stylesheets involved, to a web server from which they may be accessed, usually but not necessarily, by the public. This process is called 'Uploading'. Prior to publishing there are a few checks which should be carried out.

Setting up your site

While setting up Site Manager you may already have configured the 'remote' site, if not, either proceed as detailed there, go directly to Publish Settings via Edit > Publishing Site settings.

Confirmation of correct publication Enter the following details:

- In the 'Site names' box enter the name that you want to know the site by.
- HTTP address (URL) of your site. From your ISP (see hints).
- Publishing address - This is the ftp address to which you will publish.
- User name - From your ISP.
- Password - From your ISP.
- If you wish to, check 'Save Password'. If you have several sites set up and you have one site that you always or usually publish to you may wish to click on the name of this site then 'Set as default'. This simplifies uploading. Click OK.

Uploading

Open the page that you want to upload. An easy way to do this is from the Site Manager.

1 Click the 'Publish' button.

2 On the 'Publish Page' window on the 'Publish' tab, if it is not your default, in the 'Site name' box select the site to which you want to publish. The 'Page title' and 'File name' should already be completed.

3 If the page is to be uploaded to a sub-directory, rather than the root directory, enter the name of a sub-directory and any of the other data if required.

Note: This directory must exist. KompoZer cannot create it.

4 If it is the first time to upload the page, and if it includes images or uses external style sheets, check the box 'Include images and other files'. (If it is not the first time and these other files have not changed the box may be left unchecked.) The files will be placed in the same directory as the page. If you want them to go in a sub-directory check the box 'Use this site sub-directory' and name the directory. In this case the directory will be created if needed.

5 You should not need to refer to the 'Settings' tab as the data should be collected via the Site name you have selected but you may view the data and change if you wish.

6 Click 'Publish'. A 'Publishing' window will appear and uploading will commence. (If you are on a dial-up connection this will be connected.)

7 Within a short time you should receive confirmation of correct publication similar to the first figure above.

8 One possible source of problems occurs if you are prevented from accessing the site by a firewall. In this case you may receive a 'Publishing failed' message similar to that shown in the second figure.

The 'Troubleshooting' button takes you to the KompoZer help system but this is short of aid in this area at present.

Other possible problems include

- Some required files are missing
- File or directory names incorrect e.g. Wrong case

Once you have published a page, if you need to publish it again, your settings (e.g. subdirectories) should be remembered by KompoZer. You will not see steps 2 to 4 again unless changes have been made to the page. Fig 5 and Fig 6 shows publishing window.

Fig 5

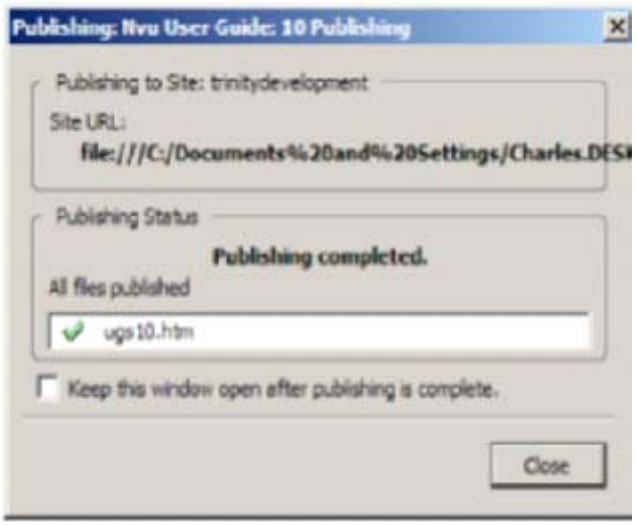


Fig 6

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;
  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>

Introduction to IIS and XAMPP, Dynamic Website and Hosting and FTP tool Filezilla - Projects in JavaScript

Objectives : At the end of this lesson you shall be able to
- **describe XAMPP**
- **describe what is included in XAMPP**
- **describe FTP**
- **describe fileZilla**
- **describe a web project**
- **follow SDLC.**

Introduction

XAMPP is a free and open source cross-platform web server solution stack package, consisting mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages.

XAMPP's name is an acronym for:

- X (to be read as "cross", meaning cross-platform)
- Apache HTTP Server
- MySQL
- PHP
- Perl

What's Included in XAMPP?

XAMPP has four primary components. These are:

1 **Apache:** Apache is the actual web server application that processes and delivers web content to a computer. Apache is the most popular web server online, powering nearly 54% of all websites.

2 **MySQL:** Every web application, howsoever simple or complicated, requires a database for storing collected data. MySQL, which is open source, is the world's most popular database management system. It powers everything from hobbyist websites to professional platforms like WordPress. You can learn how to master PHP with this free MySQL database for beginners course.

3 **PHP:** PHP stands for Hypertext Preprocessor. It is a server-side scripting language that powers some of the most popular websites in the world, including WordPress and Facebook. It is open source, relatively easy to learn, and works perfectly with MySQL, making it a popular choice for web developers.

4 **Perl:** Perl is a high-level, dynamic programming language used extensively in network programming, system admin, etc. Although less popular for web development purposes, Perl has a lot of niche applications.

Different versions of XAMPP may have additional components such as phpMyAdmin, OpenSSL, etc. to create full-fledged web servers.

XAMPP features

XAMPP requires only one zip, tar, or exe file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. XAMPP is regularly updated to incorporate the latest releases of Apache, MySQL, PHP and Perl. It also comes with a number of other modules including OpenSSL and phpMyAdmin.

Self-contained, multiple instances of XAMPP can exist on a single computer, and any given instance can be copied from one computer to another.

It is offered in both a full, standard version and a smaller version.

Use

Officially, XAMPP's designers intended it for use only as a development tool, to allow website designers and programmers to test their work on their own computers without any access to the Internet. To make this as easy as possible, many important security features are disabled by default. In practice, however, XAMPP is sometimes used to actually serve web pages on the World Wide Web. A special tool is provided to password-protect the most important parts of the package.

XAMPP also provides support for creating and manipulating databases in MySQL and SQLite among others.

Once XAMPP is installed, it is possible to treat a localhost like a remote host by connecting using an FTP client. Using a program like FileZilla has many advantages when installing a content management system (CMS) like Joomla or WordPress. It is also possible to connect to localhost via FTP with an HTML editor.

The default FTP user is "newuser", the default FTP password is "wampp". The default MySQL user is "root" while there is no default MySQL password.

XAMPP 1.8.3-4 for Windows, includes

- Apache 2.4.9
- MySQL 5.6.16
- PHP 5.5.11
- phpMyAdmin 4.1.12
- FileZilla FTP Server 0.9.41
- Tomcat 7.0.42 (with mod_proxy_ajp as connector)
- Strawberry Perl 5.16.3.1 Portable
- XAMPP Control Panel 3.2.1 (from hackattack142)

XAMPP 1.8.3-4 for Linux, includes

- Apache 2.4.9
- MySQL 5.6.16
- PHP 5.5.11
- phpMyAdmin 4.1.12
- OpenSSL 1.0.1g

XAMPP is also available for Mac OS.

File Transfer Protocol

The **File Transfer Protocol (FTP)** is a standard network protocol used to transfer computer files from one host to another host over a TCP-based network, such as the Internet.

FTP is built on a client-server architecture and uses separate control and data connections between the client and the server. FTP users may authenticate themselves using sign-in protocol, normally in the form of a username and password, but can connect anonymously if the server is configured to allow it. For secure transmission that protects the username and password, and encrypts the content, FTP is often secured with SSL/TLS (FTPS). SSH File Transfer Protocol (SFTP) is sometimes also used instead, but is technologically different.

The first FTP client applications were command-line applications developed before operating systems had graphical user interfaces, and are still shipped with most Windows, Unix, and Linux operating systems. Many FTP clients and automation utilities have since been developed for desktops, servers, mobile devices, and hardware, and FTP has been incorporated into productivity applications, such as Web page editors.

History of FTP server

The original specification for the File Transfer Protocol was written by Abhay Bhushan and published as RFC 114 on 16 April 1971. Until 1980, FTP ran on NCP, the predecessor of TCP/IP. The protocol was later replaced by a TCP/IP version, RFC 765 (June 1980) and RFC 959 (October 1985), the current specification. Several proposed standards amend RFC 959, for example RFC 2228 (June 1997) proposes security extensions and RFC 2428 (September 1998) adds support for IPv6 and defines a new type of passive mode.

Login

FTP login utilizes a normal username and password scheme for granting access. The username is sent to the server using the USER command, and the password is sent using the PASS command. If the information provided by the client is accepted by the server, the server will send a greeting to the client and the session will commence. If the server supports it, users may log in without providing login credentials, but the same server may authorize only limited access for such sessions.

Anonymous FTP

A host that provides an FTP service may provide anonymous FTP access. Users typically log into the service with an 'anonymous' (lower-case and case-sensitive in some FTP servers) account when prompted for user name. Although users are commonly asked to send their email address instead of a password, no verification is actually performed on the supplied data. Many FTP hosts whose purpose is to provide software updates will allow anonymous logins.

NAT and firewall traversal

FTP normally transfers data by having the server connect back to the client, after the PORT command is sent by the client. This is problematic for both NATs and firewalls, which do not allow connections from the Internet towards internal hosts. For NATs, an additional complication is that the representation of the IP addresses and port number in the PORT command refer to the internal host's IP address and port, rather than the public IP address and port of the NAT.

There are two approaches to this problem. One is that the FTP client and FTP server use the PASV command, which causes the data connection to be established from the FTP client to the server. This is widely used by modern FTP clients. Another approach is for the NAT to alter the values of the PORT command, using an application-level gatewayfor this purpose.

Differences from HTTP

When operating in its modern passive mode, FTP uses a single socket for both signalling and for actual file data, just like the HTTP protocol. But when used in its original configuration, in "active mode" with a separate socket for the download, FTP exhibits true out-of-band control which is not an option with HTTP.

Web browser support

Most common web browsers can retrieve files hosted on FTP servers, although they may not support protocol extensions such as FTPS. When an FTP-rather than an HTTP-URL is supplied, the accessible contents on the remote server are presented in a manner that is similar to that used for other Web content. A full-featured FTP client can be run within Firefox in the form of an extension called FireFTP.

Syntax

FTP URL syntax is described in RFC1738, taking the form: ftp://[<user>[:<password>]@]<host>[:<port>]/<url-path> The bracketed parts are optional.

For example, the URL ftp://public.ftp-servers.example.com/mydirectory/myfile.txt represents the file myfile.txt from the directory mydirectory on the server public.ftp-servers.example.com as an FTP resource. The URL ftp://user001:secretpassword@private.ftp-servers.example.com/mydirectory/myfile.txt adds a specification of the username and password that must be used to access this resource.

More details on specifying a username and password may be found in the browsers' documentation, such as, for example, Firefox and Internet Explorer. By default, most web browsers use passive (PASV) mode, which more easily traverses end-user firewalls.

Security

FTP was not designed to be a secure protocol, and has many security weaknesses. In May 1999, the authors of RFC 2577 listed a vulnerability to the following problems:

- Brute force attacks
- Bounce attacks
- Packet capture (sniffing)
- Port stealing
- Spoof attacks
- Username protection

FTP does not encrypt its traffic; all transmissions are in clear text, and usernames, passwords, commands and data can be read by anyone able to perform packet capture (sniffing) on the network. This problem is common to many of the Internet Protocol specifications (such as SMTP, Telnet, POP and IMAP) that were designed prior to the creation of encryption mechanisms such as TLS or SSL. A common solution to this problem is to use the "secure", TLS-protected versions of the insecure protocols (e.g. FTPSfor FTP, TelnetS for Telnet, etc.) or a different, more secure protocol that can handle the job, such as the SFTP/SCP tools included with most implementations of the Secure Shellprotocol.

Secure FTP

Securing FTP transfers may be accomplished by several methods.

FTPS

Explicit FTPS is an extension to the FTP standard that allows clients to request that the FTP session be encrypted. This is done by sending the "AUTH TLS" command. The server has the option of allowing or denying connections that do not request TLS. This protocol extension is defined in the proposed standard: RFC 4217. Implicit FTPS is a deprecated standard for FTP that required the use of a SSL or TLS connection. It was specified to use different ports than plain FTP.

SFTP

The SSH file transfer protocol or secure FTP (SFTP), also transfers files and has a similar command set for users, but is built on different software technology. SFTP uses theSecure Shell protocol (SSH) to transfer files. Unlike FTP, it encrypts both commands and data, preventing passwords and sensitive information from being transmitted openly over the network. It cannot interoperate with FTP software.

FTP over SSH (not SFTP)

FTP over SSH is the practice of tunneling a normal FTP session over a Secure Shell connection. Because FTP uses multiple TCP connections (unusual for a TCP/IP protocol that is still in use), it is particularly difficult to tunnel over SSH. With many SSH clients, attempting to set up a tunnel for the control channel (the initial client-to-server connection on port 21) will protect only that channel; when data is transferred, the FTP software at either end sets up new TCP connections (data channels) and thus have no confidentiality or integrity protection.

Otherwise, it is necessary for the SSH client software to have specific knowledge of the FTP protocol, to monitor and rewrite FTP control channel messages and autonomously open new packet forwardings for FTP data channels. Software packages that support this mode include:

- Tectia ConnectSecure (Win/Linux/Unix) of SSH Communications Security's software suite
- Tectia Server for IBM z/OS of SSH Communications Security's software suite
- FONC (the GPL licensed)
- Co:Z FTPSSH Proxy

Other methods of transferring files using SSH that are not related to FTP include SFTP and SCP; in each of these, the entire conversation (credentials and data) is always protected by the SSH protocol.

FILEZILLA

FileZilla is a free FTP solution. Both a client and a server are available. FileZilla is open source software distributed free of charge under the terms of the GNU General Public License. Using FileZilla files can be uploaded or downloaded from client to server and vice-versa. It is very user friendly and no commands are required to do upload and download operations. Files can be uploaded or downloaded by simple drag-drop operations.

Designing a Web Project: A project in Web should be developed after comprehensive enquiry about what exactly the client/end user want. For this some meeting can be arranged with clients to find out the exact requirement of the client. This is called SRS(System Requirement Specification).

Some methods are followed for SRS, which are giving some questions abount the system to the clients and verifying the answer to gauge the requirement of the clients. Showing them some demo screen to get their response. Collecting the reports they use to understand the type of data they use.

Before development of any system or project, SRS is very important as if you cannot understand the exact user requirement, then the system/project developed by you with lot of man hours spent on it would be totally wasted and the project would be scraped.

SDLC: The systems development life cycle (SDLC), also referred to as the application development life-cycle, is a term used in systems engineering, information systems and software engineering to describe a process for planning, creating, testing, and deploying an information system. (Fig 1)

Fig 1

The system development life cycle framework provides a sequence of activities for system designers and developers to follow. It consists of a set of steps or phases in which each phase of the SDLC uses the results of the previous one.

The SDLC adheres to important phases that are essential for developers, such as planning, analysis, design, and implementation, and are explained in the section below. It includes evaluation of present system, information gathering, feasibility study and request approval. A number of SDLC models have been created: waterfall, fountain, spiral, build and fix, rapid prototyping, incremental, and synchronize and stabilize. The oldest of these, and the best known, is the waterfall model: a sequence of stages in which the output of each stage becomes the input for the next. These stages can be characterized and divided up in different ways, including the following:

- **Preliminary analysis:** The objective of phase 1 is to conduct a preliminary analysis, propose alternative solutions, describe costs and benefits and submit a preliminary plan with recommendations.
- **Conduct the preliminary analysis:** in this step, you need to find out the organization's objectives and the nature and scope of the problem under study. Even if a problem refers only to a small segment of the organization itself then you need to find out what the objectives of the organization itself are. Then you need to see how the problem being studied fits in with them.
- **Propose alternative solutions:** In digging into the organization's objectives and specific problems, you may have already covered some solutions. Alternate proposals may come from interviewing employees, clients, suppliers, and/or consultants. You can also study what competitors are doing. With this data, you will have three choices: leave the system as is, improve it, or develop a new system.

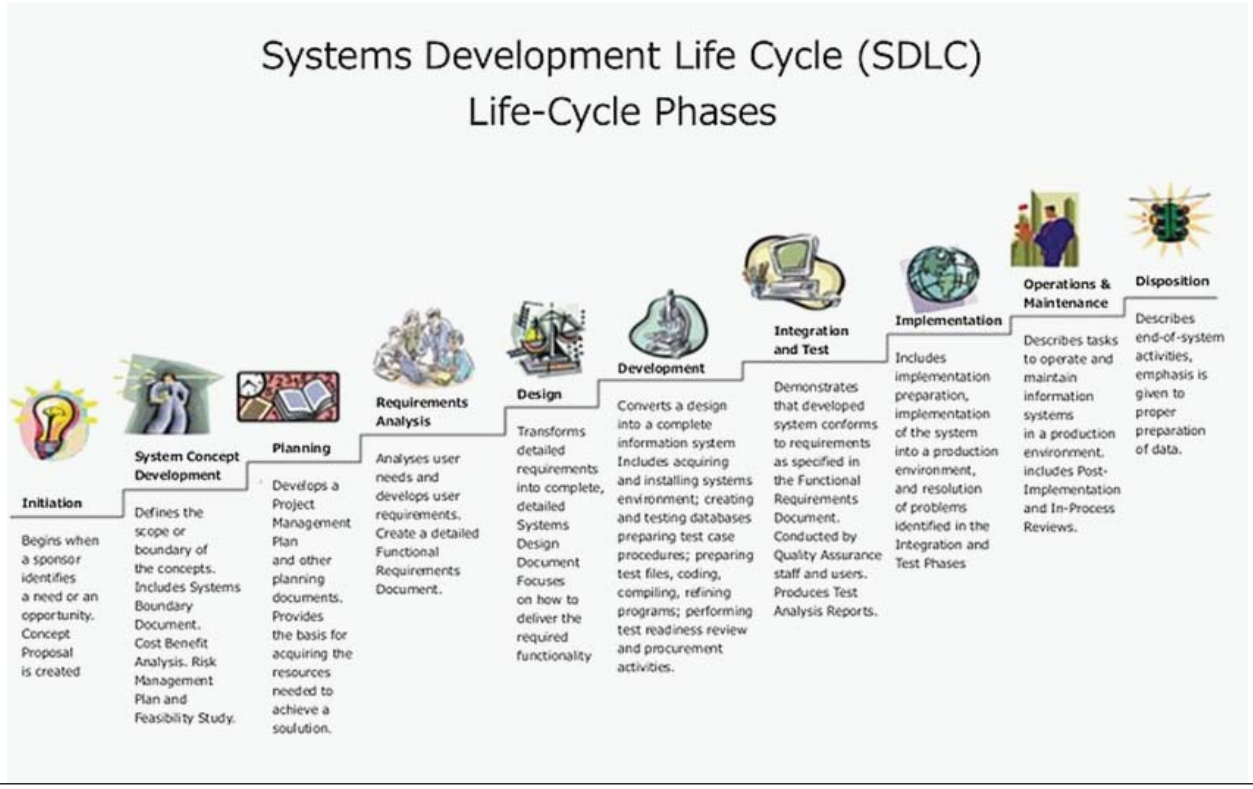
Describe the costs and benefits.

- **Systems analysis, requirements definition:** Defines project goals into defined functions and operation of the intended application. Analyzes end-user information needs.
- **Systems design:** Describes desired features and operations in detail, including screen layouts, business rules, process diagrams, pseudo code and other documentation.
- **Development:** The real code is written here.
- **Integration and testing:** Brings all the pieces together into a special testing environment, then checks for errors, bugs and interoperability.
- **Acceptance, installation, deployment:** The final stage of initial development, where the software is put into production and runs actual business.
- **Maintenance:** During the maintenance stage of the SDLC, the system is assessed to ensure it does not become obsolete. This is also where changes are made to initial software. It involves continuous evaluation of the system in terms of its performance.

- **Evaluation:** Some companies do not view this as an official stage of the SDLC, but is it an important part of the life cycle. Evaluation step is an extension of the Maintenance stage, and may be referred to in some circles as Post-implementation Review. This is where the system that was developed, as well as the entire process, is evaluated. Some of the questions that need to be answered include: does the newly implemented system meet the initial business requirements and objectives? Is the system reliable and fault-tolerant? Does the system function according to the approved functional requirements? In addition to evaluating the software that was released, it is important to assess the effectiveness of the development process. If there are any aspects of the entire process, or certain stages, that management is not satisfied with, this is the time to improve. Evaluation and assessment is a difficult issue. However, the company must reflect on the process and address weaknesses.
- **Disposal:** In this phase, plans are developed for discarding system information, hardware and software in making the transition to a new system. The purpose here is to properly move, archive, discard or destroy information, hardware and software that is being replaced, in a matter that prevents any possibility of unauthorized disclosure of sensitive data. The disposal activities ensure proper migration to a new system. Particular emphasis is given to proper preservation and archival of data processed by the previous system. All of this should be done in accordance with the organization's security requirements.

In the following example (Fig 2) these stages of the systems development life cycle are divided in ten steps from definition to creation and modification of IT work products:

Fig 2



Introduction to VBA features and applications

Objectives : At the end of this lesson you shall be able to
- **describe the applications of VBA**
- **explain the terms objects and properties**
- **explain macros, procedures and events**
- **explain the debugging techniques in VBA.**

Introduction to VBA

VBA is the acronym for Visual Basic for Applications. It is an integration of Microsoft's event-driven programming language Visual Basic with Microsoft Office applications. This is the standard Macro language used in most Microsoft Office products like Excel, Access etc. The VBA language is a derivative of Visual Basic (VB). The fundamental difference with VBA from VB is that VBA is used within an Application. By running VBA within the Microsoft Office applications, you can build customized solutions and programs to enhance the capabilities of those applications. There is built-in Visual Basic Editor in Microsoft Excel that can be used to customize and extend capabilities of MS Excel. The applications built with MS Excel are called Visual Basic for Applications, or simply VBA.

Common Uses of VBA

- Customizing and extending the functionality of the Application in which it is used.
- Automating a task you perform frequently Ex. Monthly reports etc.
- Automating repetitive operations Ex. repeating a set of actions on many workbooks
- Creating a custom command for ex. one that combines many commands to make the work faster
- Creating user interactive forms and controls like buttons to which macros or code can be assigned
- Customizing the Quick Access Toolbar with your own buttons that execute the macros you write.
- Developing new worksheet functions that can greatly simplify your formulas.
- Creating complete, macro-driven applications.
- Creating custom add-ins for Excel

Some of the common applications of VBA are:

- Keeping lists of things such as employees' details, customers'records, students' grades etc.
- Customized data entry with subsequent actions included
- Budgeting and forecasting
- Data Analysis
- Creating invoices and other forms
- Developing charts from data

Using VBA results in the following advantages for the user:

- Time Saving.
- Consistent results at much greater speed.
- Customized tasks creation to make working easy for the users.
- Functionally rich and extremely flexible.
- Simple to use and create applications

Disadvantages of VBA

As macros are the heart of VBA, there is always a threat of macro viruses embedded in them. Unless proper steps are taken to prevent this misuse, the application may be targeted in a potentially unsafe manner.This can be tackled by using proper antivirus products and verifying the certificates of the embedded macros.

Common Terms used in Excel VBA

Object oriented programming

The word Object is used to describe just about everything in Excel. In Excel, an Object can be Form, Button, Chart or even the Visual Basic Editor (VBE) itself.

In object-oriented programming, a class is a code element that defines an object. A good analogy for a class module is the specification that defines a Window, Button or a Sheet etc. You create an object using the class as its specification. A Class is therefore a template from which objects are created. Objects can be treated as instances of Classes. You add code to the class module to define the object's properties and methods. Modifying the code in a class module modifies how the object defined by the class module behaves.

In the Object Hierarchy, at the top is the Application Object, ie. Excel here. Next in this order is Workbook Object e.g.; Book1.xlsx. Directly underneath the Workbook Object comes the Worksheet Object. At the Worksheet Object the Object Hierarchy branches off to incorporate all Objects of the Worksheet. The first one you will most likely encounter will be the Range Object. Branching from the

Range Object there are again many other objects such as the Areas Object, the Borders Object, Font Object, Characters, etc. In brief, the hierarchy is as follows:

- Application
- Workbook
- Worksheet
- Range

When you have a group of Objects that are related, this is then known as a "Collection". So when we use the term Workbooks, we are referring to all open Workbooks, and when we use the term Workbook, we are only referring to an individual Workbook (the active Workbook).

Properties

Properties are attributes of an Object. They are used to define an Objects characteristics. For example; the Worksheet Object has many Properties, one of which would be its name. By changing the Visible Property, we may hide or unhide it. To be able to change the Property of any Object we must first identify the Object whose Property we wish to change. Examples of properties are height, width, font color, name etc. to name a few.

Macros

A macro is a set of commands bundled together under one name. These logically pre-recorded commands can be re executed at any later stage to repeat the task they were designed for. Macros simplify the work by eliminating rewriting the code or repeating the same steps in case of frequently needed actions. The advantage with macros is that they can be designed even by a person not knowing much of programming. The macro recording tool provided in all Office applications facilitates this. All that is needed is specifying a name and recording the steps in a sequence. A person with a knowledge of programming on the other hand can either write a macro code directly, or even edit code of the existing macros. In VBA, macros can be assigned to controls like buttons to make the application user friendly.

Procedures

Procedures are a named set of statements that are executed as a whole. They tell Excel how to perform a specific task. The task performed can be very simple or very complicated. It is good practice to break long or complicated procedures into smaller sized logically ordered procedures. The two main types of Procedures are Sub and Function. In VBA a Sub procedure either results from recording a macro or results in the creation of a macro, while Function procedure must be written only manually. All Sub procedures must begin with the word Sub followed by a chosen name and then a set of parenthesis. All Subs must end with End Sub.

Methods

A Method is simply a procedure that acts on an Object. It makes the Object do something, like opening a Workbook or deleting a Worksheet etc. Examples of some methods of VBA objects are the Activate, Copy, Delete, Save methods of the Worksheet, the Add Item and clear methods of the combo Box and list box etc.

Events

An Event in Excel VBA is the reason or trigger for an action to take place. For example a mouse click is an event, Double Clicking an object is an event, closing or opening of a Workbook are also events. The action to be taken when the event occurs is written as code in the event procedure. Examples of events are the Click, DoubleClick events of Buttons, Scroll, and Change Events of the Scroll Bar etc.

Modules

Modules are containers for holding VBA code. Modules can contain declarations and procedures. VBA code that is placed in one or more modules can be called from an Office application to perform a specified task.

The following are the elements of the VBA Module

1 Object navigation box-Used to select the object to work with

2 Declarations/Procedure navigation box-Used to navigate to the general declarations section or to a particular procedure

3 Declarations-Contains the declarations for the module

4 Procedures-Contains the sub procedures and functions for the module

There are two types of modules: standard modules and class modules.

Standard modules are modules that contain procedures that are not associated with any particular object.

Class modules are modules that are associated with a particular object. Class modules can be used either for form or report modules or for custom objects accessible throughout your VBA application. For example, form and report modules can contain code that corresponds to a particular form or report. An example of a form module is an event procedure, such as the Click event for a control on a form.

One way to create a module is to select the Modules tab in the database window and click the New button. Another way to create a module is to select Insert → Module.

The Visual Basic Editor

The Visual Basic Editor is where the actual programming is done. There are many ways to access the VBA editor.

One among them is using the shortcut key Alt + F11. You can also access the Visual Basic Editor from Tools > Macro > Visual Basic Editor from the Excel Worksheet.

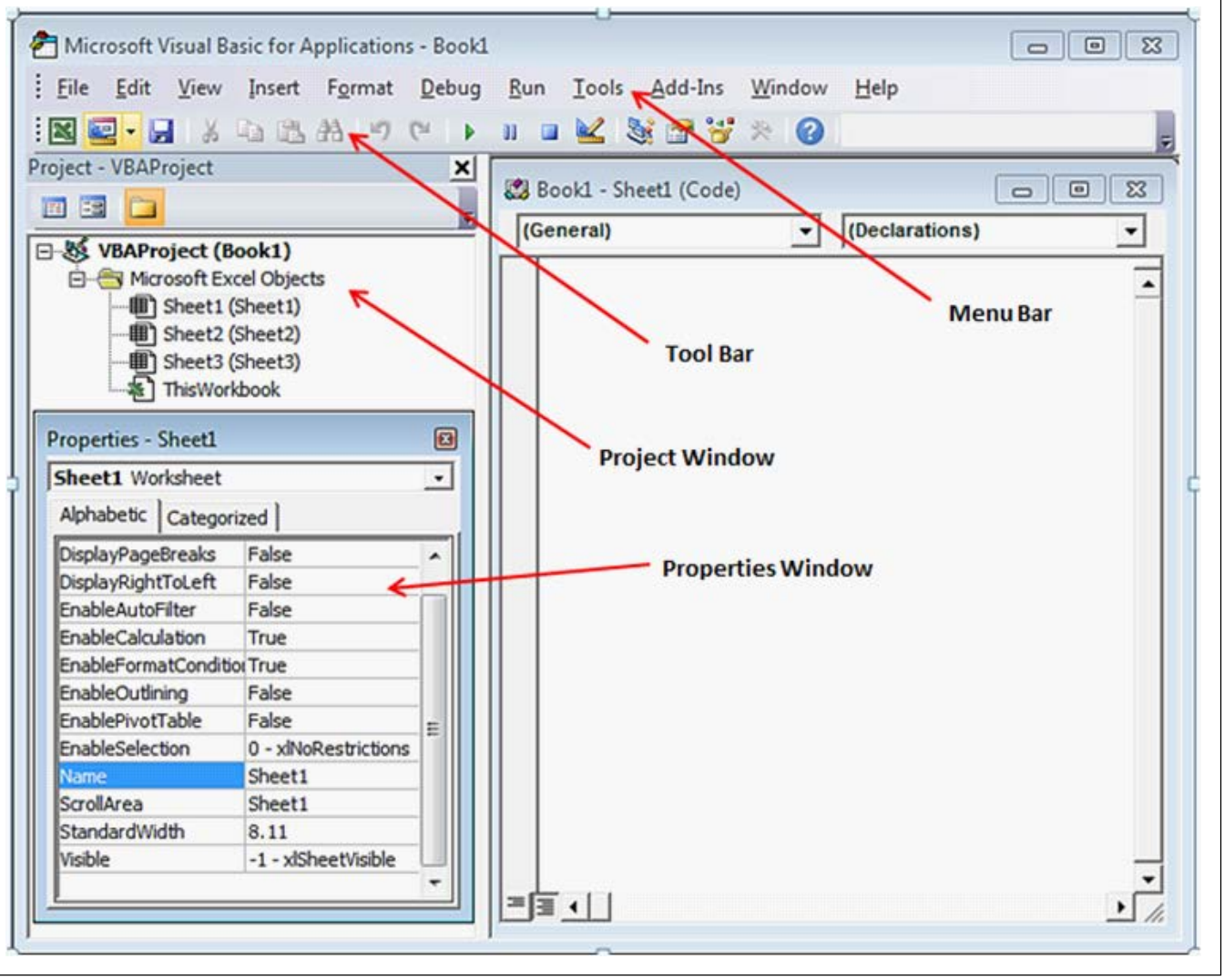
At the very top of the VBE you have what is known as the Menu Bar. From this one menu bar it is possible to access most functions of the VBE. If you right click on the grey area just to the right of Help on the Menu Bar a shortcut menu will appear. Select Customize... Here you will see the names of all the Toolbars that can be available to you. As a general default you will have the Menu Bar and the Standard Toolbar displayed.

Project Explorer and Properties Window

To open the Project Explorer, go to View > Project Explorer, or use the shortcut keys, Ctrl + R. To view the Properties window, go to View> Properties Window,or click F4. The Project explorer contains all the objects like sheets, Books, modules etc. If you now click on Sheet1 in the Project Explorer you will see in the Properties Window a list of all the Properties for a

To the right side of the project explorer and properties window is the code window where the actual programming is done. (Fig 1)

Fig 1



Saving Excel files containing macros or VBA code.

All Excel files containing VBA code or macros must be saved as type "Excel Macro Enabled WorkBook.xlsm" for the macros to be preserved.

You may have to sometimes "Enable Macros" on reopening the files containing Macros when prompted else the code will not be effective.

Debugging in VBA

Debugging is one of the most important skills for a developer. Software development is all about writing code, spotting the mistakes, and fixing them.

VBA offers very powerful debugging tools during development, with the ability to add error handling routines to help debug deployed/remote applications.

Debugging is used for fixing Bugs, analysis during development and supporting deployed applications.

Debugger

There are several parts of the debugger that work together to let you analyze how your code runs:

- Integrated Development Environment (IDE)
- Breakpoints
- Stepping Through and Over Code

Integrated Development Environment (IDE)

From the IDE, there are several things you can do:

Current Definition [Shift F2]

Put the cursor on the variable, procedure, or property in question and press [Shift F2] to see where it's defined. You'll jump directly to it. You can do this as often as you like to understand how your code works. Press [Ctrl Shift F2] to go back to where you came.

Run the Current Procedure

Highlight the procedure you want to run and press [F5] to run it. If you want to step into it line-by-line, press [F8]. Of course, running a procedure this way only works if you don't have to pass parameters to it. If you need to, consider using the Immediate Window.

Breakpoints

Breakpoints are placed on the lines in your code so the debugger is invoked when the program tries to execute that line. A breakpoint can be placed on any line that is actually run (not lines in the General Declarations section, or lines that define variables in a procedure). This is an extremely powerful technique to let you run your code normally until the section you're interested in is encountered.

Breakpoints can be added by moving to the line desired and pressing F9, clicking with the mouse on the left border, or from the Debug menu. Multiple breakpoints can be added during your debugging session. Breakpoints are temporary and are automatically removed when you close the VBA.

Stepping Through Code

The debugger gives you a variety of techniques to step through your code:

Step Into [F8]

Run the current line and go to the next one.

Step Over [Shift F8]

Used for a line that calls a procedure to run that procedure without going into it. The command lets you run the procedure (and any procedures it may call), and go to the next line in the calling procedure.

Step Out [Ctrl Shift F8]

Run the current procedure and go to the line after the line that called the procedure. This is basically a way to simplify the debugging process by letting you skip the remainder of the current procedure once you realize you don't need to step into it any more.

Set Next Statement [Ctrl F9]

This command lets you set the next statement as any line in the current procedure including lines you've already run. It is particularly useful if you run though some code and then decide you should repeat it because you missed something.

Show Next Statement

Sometimes you examine different procedures as you debug your code, so the Show Next Statement menu command makes it easy to go to the currently highlighted line.

Debugging Views

In addition to seeing which line of code runs and evaluating variables as you debug, there are several other views that help you diagnose your development environment:

- Call Stack
- Immediate Window
- Locals Window
- Watch Window

Call Stack [Ctrl L]

The call stack keeps track of the procedure calling chain so you can easily see how you got to the current procedure through all the other procedures. Retrieve it under View, Call Stack, or press [Ctrl L].

From this dialog, you can click on any procedure and jump immediately to it. Before analyzing the details of the current procedure, it may be more important to understand how and why you got there since the problem may be there rather than in the current procedure.

Immediate Window [Ctrl G]

This is the most basic debugging area. You can use the Immediate Window whether your code is running or not. Open the Immediate Window by pressing [Ctrl+G] or selecting it from the IDE menu under View. The Immediate window lets you:

- Evaluate expressions unrelated to your code (e.g. math equations)
- Evaluate variables or expressions in your code (e.g. a current variable value)
- Run code

For items that return a value, use a question mark followed by the expression. For instance:? 14/5 then click Enter to see the value. If your code is currently running and stopped, you can use this method to evaluate the current value of a variable: ? pay

Locals Window

You can see all the local variables by selecting Locals Window from the Views menu. This displays the entire list of local variables and their current values. Local variables are variables defined in the current procedure and module declaration section.

You can modify the value held by a variable by clicking on the Value column and editing it. This is an alternative to modifying values from the Immediate Window.

Watch Window

The Watch Window is similar to the Locals Window, but you specify the variables you want to track. You can track variables across modules and procedures and keep them in your Watch Window to see their value no matter where the current line is.

The first step is to add a variable to the Watch Window. This can be done by placing the cursor in the variable you want to track, and selecting Debug, Add Watch, or from the right-click menu, selecting Add Watch. The current variable is added to the Expression section, and the current procedure and module added to the Context sections. You can also add expressions, and options to break when the value changes. This is particularly useful when you are having trouble determining why a particular situation arises in your application.

Form Controls and Properties in VBA

Objectives: At the end of this lesson you shall be able to
- **differentiate form controls and activeX controls in VBA**
- **describe the types of form controls and activeX controls**
- **describe the properties, methods and events of controls.**

Introduction

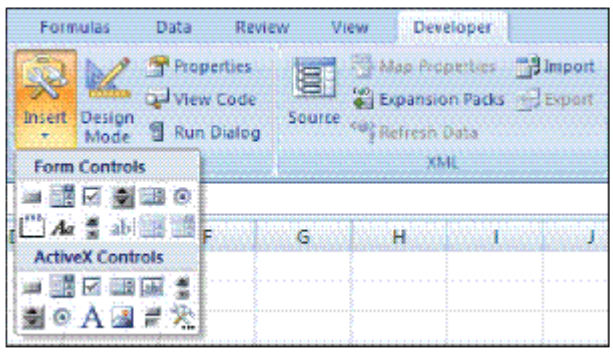
Controls are objects that display data or make it easier for users to enter or edit data, perform an action, or make a selection. In general, controls make the form easier to use. Examples of common controls include list boxes, option buttons, and command buttons. Controls can also run assigned macros and respond to events, such as mouse clicks, by running Visual Basic for Applications (VBA) code. Events associated with the controls can be used to run a specific code to do a specific job.

For added flexibility, you can add controls and other drawing objects to the worksheet, and combine and coordinate them with worksheet cells. For example, you can use a list box control to make it easier for a user to select from a list of items. Or, you can use a spin button control to make it easier for a user to enter a number.

You can display or view controls and objects alongside associated text that is independent of row and column boundaries without changing the layout of a grid or table of data on your worksheet. Many of these controls can also be linked to cells on the worksheet and do not require VBA code to make them work. You can set properties that determine whether a control floats freely or moves and resizes together with a cell. For example, you might have a check box that you want to move together with its underlying cell when the range is sorted.

Excel has two types of controls: Form controls and ActiveX Controls (Refer Fig 1).

Fig 1



Form controls are built in to Excel whereas ActiveX controls are loaded from separate Dynamic Link Libraries.

ActiveX controls provide more flexibility in changing or setting the properties both at design time and run time. Form controls allow you to assign predefined macros or subroutines to them but offer limited choices in modifying the properties. In addition to these sets of controls, you can also add objects from the Drawing tools, such as Auto Shapes, Word Art, Smart Art graphic, or text boxes.

Form controls

You use Form controls when you want to easily reference and interact with cell data without using VBA code, and when you want to add controls to chart sheets. For example, after you add a list box control to a worksheet and linking it to a cell, you can return a numeric value for the current position of the selected item in the control. You can then use that numeric value in conjunction with the INDEX function to select different items from the list.

You can also run macros by using Form controls. You can attach an existing macro to a control, or write or record a new macro. When a user of the form clicks the control, the control runs the macro.

However, these controls cannot be added to UserForms, used to control events, or modified to run Web scripts on Web pages. The Summary of form controls is given in Table 1.

ActiveX controls

ActiveX controls can be used on worksheet forms, with or without the use of VBA code, and on VBA UserForms. In general, use ActiveX controls when you need more flexible design requirements than those provided by Form controls. ActiveX controls have extensive properties that you can use to customize their appearance, behavior, fonts, and other characteristics.

You can also control different events that occur when an ActiveX control is interacted with. For example, you can perform different actions, depending on which choice a user selects from a list box control, or you can query a database to refill a combo box with items when a user clicks a button. You can also write macros that respond to events associated with ActiveX controls. When a user of the form interacts with the control, your VBA code then runs to process any events that occur for that control.

Table 1

Control Name	Example	Description
	Label	Identifies the purpose of a cell or text box, or displays descriptive text (such as titles, captions, pictures) or brief instructions.
	Group box	Groups related controls into one visual unit in a rectangle with an optional label. Typically, option buttons, check boxes, or closely related contents are grouped.
	Button	Runs a macro that performs an action when a user clicks it. A button is also referred to as a push button.
	Check box	Turns on or off a value that indicates an opposite and unambiguous choice. You can select more than one check box on a worksheet or in a group box. A check box can have one of three states: selected (turned on), cleared (turned off), and mixed, meaning a combination of on and off states (as in a multiple selection).
	Option button	Allows a single choice within a limited set of mutually exclusive choices; an option button is usually contained in a group box or a frame. An option button can have one of three states: selected (turned on), cleared (turned off), and mixed, meaning a combination of on and off states (as in a multiple selection). An option button is also referred to as a radio button.
	List box	Displays a list of one or more items of text from which a user can choose. Use a list box for displaying large numbers of choices that vary in number or content. There are three types of list boxes:
		A single-selection list box enables only one choice. In this case, a list box resembles a group of option buttons, except that a list box can handle a large number of items more efficiently.
		A multiple-selection list box enables either one choice or contiguous (adjacent) choices.
		An extended-selection list box enables one choice, contiguous choices, and non-contiguous (or disjointed) choices.
	Combo box	Combines a text box with a list box to create a drop-down list box. A combo box is more compact than a list box but requires the user to click the down arrow to display the list of items. Use a combo box to enable a user to either type an entry or choose only one item from the list. The control displays the current value in the text box, regardless of how that value is entered.
	Scroll bar	Scrolls through a range of values when you click the scroll arrows or drag the scroll box. In addition, you can move through a page (a preset interval) of values by clicking the area between the scroll box and either of the scroll arrows. Typically, a user can also type a text value directly into an associated cell or text box.
	Spin button	Increases or decreases a value, such as a number increment, time, or date. To increase the value, click the up arrow; to decrease the value, click the down arrow. Typically, a user can also type a text value directly into an associated cell or text box.

Your computer also contains many ActiveX controls that were installed by Excel and other programs, such as Calendar Control 12.0 and Windows Media Player.

Not all ActiveX controls can be used directly on worksheets, some can be used only on Visual Basic for Applications (VBA) UserForms. If you try to add any one of these particular ActiveX controls to a worksheet, Excel displays the message "Cannot insert object."

However, Active X controls cannot be added to chart sheets from the user interface or to XLM macro sheets. You also cannot assign a macro to run directly from an ActiveX control the same way you can from a Form control. The summary of the Active X controls is given in Table 2. The properties and events of commonly used ActiveX controls are shown in Table 3.

Table 2

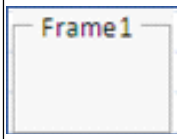
Control Name	Example	Description
	Check box	Turns on or off a value that indicates an opposite and unambiguous choice. You can select more than one check box at a time on a worksheet or in a group box. A check box can have one of three states: selected (turned on), cleared (turned off), and mixed, meaning a combination of on and off states (as in a multiple selection).
	Text box	Enables you to, in a rectangular box, view, type, or edit text or data that is bound to a cell. A text box can also be a static text field that presents read-only information.
	Command button	Runs a macro that performs an action when a user clicks it. A command button is also referred to as a push button.
	Option button	Allows a single choice within a limited set of mutually exclusive choices usually contained in a group box or frame. An option button can have one of three states: selected (turned on), cleared (turned off), and mixed, meaning a combination of on and off states (as in a multiple selection). An option button is also referred to as a radio button.
	List box	Displays a list of one or more items of text from which a user can choose. Use a list box for displaying large numbers of choices that vary in number or content. There are three types of list boxes:
		A single-selection list box enables only one choice. In this case, a list box resembles a group of option buttons, except that a list box can handle a large number of items more efficiently.
		A multiple selection list box enables either one choice or contiguous (adjacent) choices.
		An extended-selection list box enables one choice, contiguous choices, and noncontiguous (or disjointed) choices.
	Combo box	Combines a text box with a list box to create a drop-down list box. A combo box is more compact than a list box, but requires the user to click the down arrow to display the list of items. Use to allow a user to either type an entry or choose only one item from the list. The control displays the current value in the text box, regardless of how that value is entered.
	Toggle button	Indicates a state, such as Yes/No, or a mode, such as On/Off. The button alternates between an enabled and disabled state when it is clicked.
	Spin button	Increases or decreases a value, such as a number increment, time, or date. To increase the value, click the up arrow; to decrease the value, click the down arrow. Typically, a user can also type a text value into an associated cell or text box.
	Scroll bar	Scrolls through a range of values when you click the scroll arrows or drag the scroll box. In addition, you can move through a page (a preset interval) of values by clicking the area between the scroll box and either of the scroll arrows. Typically, a user can also type a text value directly into an associated cell or text box.
	Label	Identifies the purpose of a cell or text box, displays descriptive text (such as titles, captions, pictures), or provides brief instructions.
	Image	Embeds a picture, such as a bitmap, JPEG, or GIF.
	Frame control	A rectangular object with an optional label that groups related controls into one visual unit. Typically, option buttons, check boxes, or closely related contents are grouped in a frame control.
		Note The ActiveX frame control is not available in the ActiveX Controls section of the Insert command. However, you can add the control from the More Controls dialog box by selecting Microsoft Forms 2.0 Frame.
	More Controls	Displays a list of additional ActiveX controls available on your computer that you can add to a custom form, such as Calendar Control 12.0 and Windows Media Player. You can also register a custom control in this dialog box.

Table 3

Control Name	Properties in addition to the applicable properties mentioned above	Some of the common events for the control
Check box	GroupName,TripleState, Value, WordWrap etc.	Change, Click, DblClick, GotFocus, LostFocus, KeyPress etc.
Text box C	Maxlength, MultiLine, Password haracter, ScrollBars, Text, TextAlign, WordWrap etc.	Change, DBLClick, GotFocus, LostFocus, KeyPress etc
Command button	Picture,Shadow, WordWrap etc.	Click, DBLClick, GotFocus, LostFocus, KeyPress etc
Option button	GroupName,TripleState, Value, WordWrap etc.	Change, Click, DBLClick, GotFocus, LostFocus, KeyPress etc.
List box	BoundColumn, ColumnCount, LinkedCell, ListFillRange, ListStyle, MatchEntry, Multiselect etc. Methods: AddItem, Clear, RemoveItem	Change, Click, DBLClick, GotFocus, LostFocus etc.
Combo box	BoundColumn, ColumnCount, LinkedCell, ListFillRange, ListStyle, MatchEntry, Matchrequired, Multiselect, Style etc. Methods: AddItem, Clear, RemoveItem	Change, Click, DBLClick, GotFocus, LostFocus etc.
Toggle button	TripleState, Value etc	Change, Click, DBLClick, GotFocus, LostFocus etc.
Spin button	Delay, LargeChange, Max, Min, Orientation, SmallChange, Value etc.	Change, SpinDown, SpinUp, GotFocus, LostFocus, KeyPress etc.
Scroll bar	Delay, LargeChange, Max, Min, Orientation, SmallChange, Value etc.	Change, Scroll, GotFocus, LostFocus, KeyPress etc.
Label	Caption, SpecialEffect, TextAlign, WordWrap etc.	Click, DBLClick, GotFocus, LostFocus etc.
Image	Picture, PictureSizeMode, PictureTiling, SpecialEffect etc. Methods: LoadPicture(RunTime)	Click, DBLClick, GotFocus, LostFocus etc.
Frame control	Cycle, KeepScrollBarsVisible, PictureSizeMode, PictureTiling, Scrollbars, Zoom etc.	RedoAction, UndoAction, Repaint, Scroll, Set Default Tab Order

The 'Design Time Properties" common to most of the controls are Name, Autoload, Backcolor, BackStyle, BorderColor, BorderStyle, Enabled, Font, ForeColor, Height, Left, Top, Visible, Width etc.

The Methods common to most of the controls are Activate, Copy, Cut, Delete, Select, Update etc.

In addition to these Methods certain functions / methods allow setting the properties at "Run Time", ie. through code.

Workbook and worksheet objects

Objectives: At the end of this lesson you shall be able to
- **describe collections in VBA**
- **describe the properties, events and methods of the workbook object**
- **describe the properties, events and methods of the worksheet ojbect.**

Collections

A collection is a series of items where each item has the same characteristics. In other words, all items can be described the same way. Programmatically, a collection is a series of items where all items share the same properties and methods, if any. For example, if you want to loop through all worksheets in a workbook, you can refer worksheets collection of the workbook and use the worksheet. Collections are a powerful feature in VBA. One of the most used, functions of Collections is their ability to provide you with a unique list.

Collections can hold a lot of data with one variable. The data in a Collection does not have to be the same type, like using a Variant array. Also, you don't have to allocate memory for Collections like you do for arrays. You simply add items to the collection and the memory is allocated dynamically.

A Collection object has four methods (and no properties). They are Add, Count, Item, and Remove. Charts, Workbooks, Shapes are some of the collections in VBA.

Workbook Object

The Workbook object is a member of the Workbooks collection. The Workbooks collection contains all the Workbook objects currently open in Microsoft Excel.

'This' Workbook Property

The ThisWorkbook property returns the workbook where the Visual Basic code is running. In most cases, this is the same as the active workbook. However, if the Visual Basic code is part of an add-in, the ThisWorkbook property won't return the active workbook. In this case, the active workbook is the workbook calling the add-in, whereas the ThisWorkbook property returns the add-in workbook. If you'll be creating an add-in from your Visual Basic code, you should use the ThisWorkbook property to qualify any statement that must be run on the workbook you compile into the add-in.

Some of the properties of Workbook are shown in Table 1.

Table 1

Full Name	Returns the name of the object, including its path on disk, as a string. Read-only String.
Name	Returns a String value that represents the name of the object.
Path	Returns a String that represents the complete path to the workbook/file that this workbook object respresents.
Worksheets	Returns a Sheets collection that represents all the worksheets in the specified workbook. Read-only Sheets object.

Some of the methods of Work Book are shown in Table 2

Table 2

Activate	Activates the first window associated with the workbook.
Close	Closes the object.
Protect	Protects a workbook so that it cannot be modified.
Save	Saves changes to the specified workbook.
SaveAs	Saves changes to the workbook in a different file.

Some of the events of WorkBook are shown in Table 3.

Table 3

Activate	Occurs when a workbook, worksheet, chart sheet, or embedded chart is activated.
Open	Occurs when the workbook is opened.

Worksheet Object

The Worksheet object is a member of the Worksheets collection. The Worksheets collection contains all the Worksheet objects in a workbook.

The Worksheet object is also a member of the Sheets collection. The Sheets collection contains all the sheets in the workbook (both chart sheets and worksheets).Some of the properties of WorkSheet are shown in Table 4.

Table 4

Application	When used without an object qualifier, this property returns an Application object that represents the Microsoft Excel application. When used with an object qualifier, this property returns an Application object that represents the creator of the specified object (you can use this property with an OLE Automation object to return the application of that object). Read-only.
Cells	Returns a Range object that represents all the cells on the worksheet (not just the cells that are currently in use).
Columns	Returns a Range object that represents all the columns on the active worksheet. If the active document isn't a worksheet, the Columns property fails.
Range	Returns a Range object that represents a cell or a range of cells.
Rows	Returns a Range object that represents all the rows on the specified worksheet. Read-only Range object.
Index	Returns a Long value that represents the index number of the object within the collection of similar objects.
Name	Returns or sets a String value that represents the object name.
Sort	Returns a Sort object. Read-only.
Visible	Returns or sets an XlSheet Visibility value that determines whether the object is visible.

Some of the methods of WorkSheet are shown in Table 5

Table 5

Activate	Makes the current sheet the active sheet.
Copy	Copies the sheet to another location in the workbook.
Delete	Deletes the object.
PrintOut	Prints the object.
SaveAs	Saves changes to the chart or worksheet in a different file.
Select	Selects the object.
ShowDataForm	Displays the data form associated with the worksheet.

Some of the events of WorkSheet are shown in Table 6

Table 6

Activate	Occurs when a workbook, worksheet, chart sheet, or embedded chart is activated.
Calculate	Occurs after the worksheet is recalculated, for the Worksheet object.
Change	Occurs when cells on the worksheet are changed by the user or by an external link.

VBA Data types, Variables and Constants

Objectives: At the end of this lesson you shall be able to
- **list the data types in VBA**
- **declare variables and assign values**
- **describe the option explicit statement.**

Introduction

Variables are entities that hold data. In VBA, variables are areas allocated by the computer memory to hold data. The following are the variable naming rules in VBA:

a Variable Names

The following are the rules when naming the variables in VBA

1 A Variable name must start with a letter and not a number. Numbers can be included within the name, but not as the first character.

2 A Variable name can be no longer than 250 characters.

3 A Variable name cannot be the same as any one of Excel's key words. For ex. you cannot name a Variable with such names as Sheet, Worksheet etc.

4 All Variables must consist of one continuous string of characters only. You can separate words by using the underscore.

b Declaring Variables

In VBA, the variables are declared before using them by assigning names and data types. Declaring variables before use tells the computer to allocate a certain memory for the variable data to be placed. Though it is a good practice to declare variables before use, in Visual Basic it is not actually compulsory to specifically declare a variable before it is used. If a variable isn't declared, VB will automatically declare the variable as a Variant. A variant is data type that can hold any type of data.

To declare a variable we use the word "Dim" (short for Dimension) followed by our chosen variable name then the word "As" followed by the variable type. For ex. Dim n as Integer.

You may also combine more variables to be declared in one line, separating each variable with a comma, as follows:

Dim first_name As String, joining_date As Date, Pay As Integer.

Declaring a variable before use is a good programming practice for the following reasons:

1 **Memory & Calculation Speed:** If you do not declare a variable to have a data type, it will, by default, have the Variant type. This takes up more memory than many of the other data types. Sometimes, Variant data types also take more time to process and at times may slow down the process.

2 **Prevention of typing errors:** If you always declare your variables, then you can use a VBA option to force you to declare variables. This will prevent you from introducing errors in your code by accidentally typing a variable name incorrectly.

3 **Highlighting wrong Data Values:** If you declare a variable to have a specific data type, and you attempt to assign the wrong type of data to it, this will generate an error in your program.

The Option Explicit Statement

The option 'Explicit' forces you to declare all variables that you use in your VBA code, by highlighting any undeclared variables as errors during compilation (before the code will run). To use this option, simply type the line as the very first line of the program (In the General Declarations section).

If you select the 'Require Variable Declaration' option of your VBA editor, the statement'Option Explicit' is always automatically included at the top of every new VBA module that is created.

To do this:

- In the Visual Basic Editor, select **Tools → Options**...
- Ensure the **Editor** tab is selected
- Check the box next to the option **Require Variable Declaration** and click OK

Keywords

Keywords in Excel VBA are words that Excel has set aside to use in the execution of code. This means we cannot use them for any other purpose. For example, Select, Active, Sub, End, Function etc are all Keywords that we can only use for their intended purpose.

Some of the reserved keywords are shown in Table 1.

Table 1

ByVal	Call	Case	CBool	CByte	CDate
CDbl	CInt	CLng	Const	CSng	CStr
Date	Dim	Do	Double	Each	Else
ElseIf	End	EndIf	Error	False	For
Function	Get	GoTo	If	Integer	Let
Lib	Long	Loop	Me	Mid	Mod
New	Next	Not	Nothing	Option	Or (Bitwise)
Or (Condition)	Private	Public	ReDim	REM	Resume
Select	Set	Single	Static	Step	String
Sub	Then	To	True	Until	vbCrLf
vbTab	With	While	Xor		

Data Types:

Visual Basic classifies data into two major categories, the numeric data types and the non-numeric data types.

Numeric data types are types of data that consist of numbers, which can be computed mathematically with various standard operators such as +, -, x, / and more. Examples of numeric data types are examination marks, height, weight, the number of students in a class, share values, price of goods, monthly bills, fees and others.

In VBA, numeric data are divided into 7 types, depending on the range of values they can store. Calculations that only involve round figures or data that does not need precision can use Integer or Long integer in the computation. Programs that require high precision calculation need to use Single and Double decision data types, they are also called floating point numbers. For currency calculation, you can use the currency data types. Lastly, if even more precision is required to perform calculations that involve many decimal points, we can use the decimal data types. These numeric data types summarized in Table 2.

Table 2

Type	Storage	Range of Values
Byte	1 byte	0 to 255
Integer	2 bytes	-32,768 to 32,767
Long	4 bytes	-2,147,483,648 to 2,147,483,648
Single	4 bytes	-3.402823E+38 to -1.401298E-45 for negative values 1.401298E-45 to 3.402823E+38 for positive values.
Double	8 bytes	-1.79769313486232e+308 to -4.94065645841247E-324 for negative values 4.94065645841247E-324 to 1.79769313486232e+308 for positive values.
Currency	8 bytes	-922,337,203,685,477.5808 to 922,337,203,685,477.5807
Decimal	12 bytes	+/- 79,228,162,514,264,337,593,543,950,335 if no decimal is use +/- 7.9228162514264337593543950335 (28 decimal places).

Non Numeric Data Types

Non-numeric data types are data that cannot be manipulated mathematically using standard arithmetic operators. The non-numeric data comprises text or string data types, the Date data types, the Boolean data types that store only two values (true or false), Object data type and Variant data type.The non numeric data types are summarized in Table 3.

Table 3

Data Type	Storage	Range
String(fixed length)	Length of string	1 to 65,400 characters
String(variable length)	Length + 10 bytes	0 to 2 billion characters
Date	8 bytes	January 1, 100 to December 31, 9999
Boolean	2 bytes	True or False
Object	4 bytes	Any embedded object
Variant(numeric)	16 bytes	Any value as large as Double
Variant(text)	Length+22 bytes	Same as variable-length string

Enumerated Data Types

If you have a set of unchanging values that are logically related to each other, you can define them together in an enumeration. This provides meaningful names for the enumeration and its members, which are easier to remember than their values. You can then use the enumeration members in many places in your code.

An enumeration has a name, an underlying data type, and a set of members. Each member represents a constant.

The Enum statement can declare the data type of an enumeration. Each member takes the enumeration's data type. You can specify Byte, Integer, Long etc..If you do not specify datatype for the enumeration, each member takes the data type of its initializer. If you specify both datatype and initializer, the data type of initializer must be convertible to data type. If neither datatype nor initializer is present, the data type defaults to Integer.

Ex.

```
Public Enum OS
Windows
Linux
Unix
DOS
MAC
End Enum
```

Suffixes for Literals

Literals are values that you assign to data. In some cases, we need to add a suffix to a literal so that VB can handle the calculation more accurately. For example, we can use pay=12000@ for a Currency type data. Some of the suffixes are displayed in Table 4.

Table 4

Suffix	Data Type
&	Long
!	Single
#	Double
@	Currency

Note: Enclose string literals within two quotations

Enclose date and time literals within two # symbols. Ex:

```
Dim marks%          'integer
Dim gorss_pay&      'long
Dim Average!        'single
Dim Total#          'double
Dim Profit@         'currency
Dim FirstName$      'string
```

If the data type is not specified, VB will automatically declare the variable as a Variant.

Named Constants

If you have a value that never changes in your application, you can define a named constant and use it in place of a literal value. A name is easier to remember than a value. You can define the constant just once and use it in many places in your code. If in a later version you need to redefine the value, the Const statement is the only place you need to make a change.

You can use Const only at module or procedure level. This means the declaration context for a variable must be a class, structure, module, procedure, or block, and cannot be a source file, namespace, or interface

Example:Const Pi As Single=3.142

Assigning Values to Variables

After declaring various variables using the Dim keywords or other keywords, we need to assign values or information to those variables. Assigning a value to a variable means storing the value in that variable. The form of an assignment statement is as follows:

Variable = Expression

The variable can be a declared variable or a control's property value. The expression could be a mathematical expression, a number, a literal value, a string, a Boolean value (true or false) , a combination of other variables and constants, a function and more. The following are some examples:

Basic = 10000

DA = Basic * 0.9

First Name = "Uma"

Label1.Caption = "Enter your age"

Command 1 Visible = false

Textbox.multiline = True

Label 1 Caption = textbox1.Text

A type mismatch error occurs when you try to assign a value to a variable of incompatible data type.

Operators in VBA and operator precedence

Objectives : At the end of this lesson you shall be able to
• **explain the various operators and their precedence in VBA.**

Operators in VBA

An **Operator** can be defined using a simple expression - 4 + 5 is equal to 9. Here, 4 and 5 are called **operands** and + is called **operator**. VBA supports following types of operators "

- Arithmetic Operators
- Comparison Operators
- Logical (or Relational) Operators
- Concatenation Operators

The Arithmetic Operators

Following arithmetic operators are supported by VBA.

Assume variable A holds 5 and variable B holds 10, the results of the various operators as shown in Table 1.

Table 1

Operator	Description	Example
+	Adds the two operands	A + B will give 15
-	Subtracts the second operand from the first	A - B will give -5
*	Multiplies both the operands	A * B will give 50
/	Divides the numerator by the denominator	B / A will give 2
%	Modulus operator and the remainder after an integer division	B % A will give 0
^	Exponentiation operator	B ^ A will give 100000

The Comparison Operators

There are following comparison operators supported by VBA.

Table 1 Assume variable A holds 10 and variable B holds 20, the results of various comparison operators as shown

Table 2

Operator	Description	Example
=	Checks if the value of the two operands are equal or not. If yes, then the condition is true	(A = B) is False
<>	Checks if the value of the two operands are equal or not. If the values are not equal, then the condition is true	(A <> B) is True
>	Checks if the value of the left operand is greater than the value of the right operand. If yes, then the condition is true	(A > B) is False
<	Checks if the value of the left operand is less than the value of the right operand. If yes, then the condition is true	(A < B) is True
>=	Checks if the value of the left operand is greater than or equal to the value of the right operand. If yes, then the condition is true	(A >= B) is False
<=	Checks if the value of the left operand is less than or equal to the value of the right operand. If yes, then the condition is true	(A <= B) is True

The Logical Operators

Following logical operators are supported by VBA.

Assume variable A holds 10 and variable B holds 0, the results on the various logical operators shown in Table 3

Table 3

Operator	Description	Example
AND	Called Logical AND operator. If both the conditions are True, then the Expression is true	a<>0 AND b<>0 is False
OR	Called Logical OR Operator. If any of the two conditions are True, then the condition is true	a<>0 OR b<>0 is true
NOT	Called Logical NOT Operator. Used to reverse the logical state of its operand. If a condition is true, then Logical NOT operator will make false	NOT(a<>0 OR b<>0) is false
XOR	Called Logical Exclusion. It is the combination of NOT and OR Operator. If one, and only one, of the expressions evaluates to be True, the result is True.	(a<>0 XOR b<>0) is true

The Concatenation Operators

Following Concatenation operators are supported by VBA.

Assume variable A holds 5 and variable B holds 10, the result of various concatenation operators shown in Table 4

Table 4

Operator	Description	Example
+	Adds two Values as Variable. Values are Numeric	A + B will give 15
&	Concatenates two Values	A & B will give 510

Assume variable A = “Microsoft” and variable B = “VBScript”, the result of the various concatenation shown in Table 5

Table 5

Operator	Description	Example
+	Concatenates two Values	A + B will give MicrosoftVBScript
&	Concatenates two Values	A & B will give MicrosoftVBScript

Note: Concatenation Operators can be used for both numbers and strings. The output depends on the context, if the variables hold numeric value or string value.

Precedence

When several operations occur in an expression, each part is evaluated and resolved in a predetermined order called operator precedence.

When operators have the same precedence they are evaluated from left-to-right.Parentheses can be used to override the order and to evaluate certain parts of the expression. Operations inside parentheses are always performed before those outside.

When a series of operators appear in the same expression there is a strict order in which they will be evaluated.

The rules of precedence tell the compiler which operators to evaluate first.

Parentheses can obviously be used to change the order of precedence.

Operators are evaluated in the following order: Mathematical, Concatenation, Relational, Logical.

The table 6 shows the precedence order of operators.

Table 6

Order	Operator	Symbol
1	Exponentiation	^
2	Negation	-
3	Multiplication	*
3	Division	/
4	Division with Integer result	\
5	Modulo	Mod
6	Addition	+
6	Subtraction	-
7	String Concatenation	&
8	Equal or Assignment	=
8	Not Equal To	<>
8	Less Than	<
8	Greater Than	>
8	Less Than or Equal To	<=
8	Greater Than or Equal To	>=
9	Not	NOT
10	And	AND
11	Or	OR
12	Exclusive OR	XOR
13	Equivalence	EQV
14	Implication	IMP

The table 7 shows the expression, steps to evaluate and the result.

Expression	First Step	Second Step	Third Step	Result
3^(15/5)*2-5	3^3*2-5	27*2-5	54-5	49
3^((15/5)*2-5)	3^(3*2-5)	3^(6-5)	3^1	3
3^(15/(5*2-5))	3^(15/(10-5))	3^(15/5)	3^3	27

VBA Message boxes and Input boxes

Objectives: At the end of this lesson you shall be able to
- **state the uses of message boxes and input boxes in VBA**
- **describe the msgbox method and msgbox function**
- **describe the inputbox method and inputbox function.**

Introduction

Many applications depend on data input from users to take the necessary action. Excel VBA has very useful functions that allow you to gather user input for your applications. VBA allows you to create message boxes, user input forms and input boxes to get user input.VBA message boxes provide a way to give information to a user and get information from a user while the program is running. The input Box function can be used to prompt the user to enter a value.

Message Box

In VBA Message Boxes fall into two basic categories, the MsgBox method and the MsgBox function.

The MsgBox Method

The message box method is used to display a pre- defined message to the user. It also contains a single command button "OK" to allow the user to dismiss the message and they must do so before they can continue working in the program.

The basic form of the Message Box (msgbox) in VBAis :Msgbox("message")

Example:

```
Sub result()

Msgbox("congratulations")

End sub
```

This displays a message box as shown in Fig 1

Fig 1

Customize the buttons in a VBA message box

The Msgbox() can be customized by changing the buttons and icons placed on it.

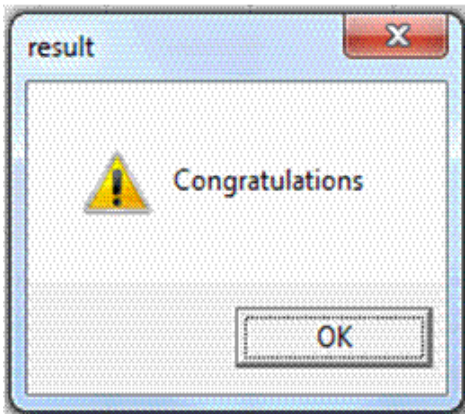
A list of various buttons and icons that can be used in the VBA message box is shown in the Table 1.

For ex. to add an icon and a title to the Msgbox() we can write the following code

```
Sub test()

Dim n As Integer

n = MsgBox("Congratulations", vbExclamation, "result")

End Sub
```

This will produce the following result as in Fig 2.

Fig 2



The MsgBox Function

The MsgBox Function displays a message in a dialog box, waits for the user to click a button, and then returns an integer indicating which button was clicked by the user.The syntax of the Msgbox() function is :

Return value = MsgBox(Prompt, Button and Icon types, Title, Help File, Help File Context)

Table 1

Constant	Description
vbOKOnly	It displays a single OK button
vbOKCancel	It displays two buttons OK and Cancel.
vbAbortRetryIgnore	It displays three buttons Abort, Retry, and Ignore.
vbYesNoCancel	It displays three buttons Yes, No, and Cancel.
vbYesNo	It displays two buttons Yes and No.
vbRetryCancel	It displays two buttons Retry and Cancel.
vbCritical	It displays a Critical Message icon.
vbQuestion	It displays a Query icon.
vbExclamation	It displays a Warning Message icon.
vbInformation	It displays an Information Message icon.
vbDefaultButton1	First button is treated as default.
vbDefaultButton2	Second button is treated as default.
vbDefaultButton3	Third button is treated as default.
vbDefaultButton4	Fourth button is treated as default.
vbApplicationModal	This suspends the current application till the user responds to the message box.
vbSystemModal	This suspends all the applications till the user responds to the message box.
vbMsgBoxHelpButton	This adds a Help button to the message box.
VbMsgBoxSetForeground	Ensures that message box window is foreground.
vbMsgBoxRight	This sets the Text to right aligned
vbMsgBoxRtlReading	This option specifies that text should appear as right-to-left.

Where:

Return Value: Indicates the action the user took when the message box was shown to him/her.

Prompt : It is the message contained in the main body of the message box.

Button and Icon Types : This specifies the set of buttons & Icons and their placement as they would appear to the user.

Help File : This is the path to a help file that the user can refer to on this topic.

Help File Context : This is the pointer to that part of the help file that specifically deals with this message.

Values returned by MsgBox Function:

VBA MsgBox function returns a value based on the user input. These values can be anyone of the ones shown in Table 2.

A Msgbox function example is shown in the code mentioned below.

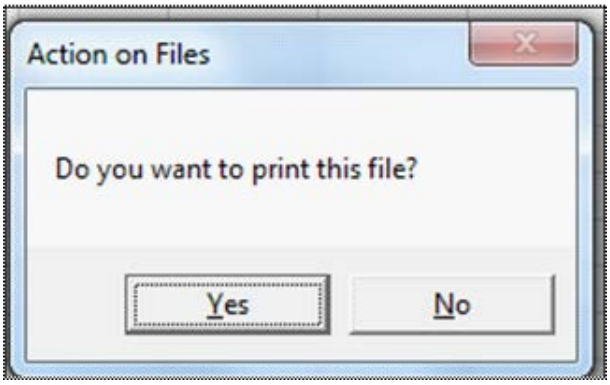
```
Sub test()

Dim n As Integer

 n = MsgBox("Do you want to print this file?", vbYesNo,
"Action on Files")

End Sub
```

Table 2

Value	Description
1	Specifies that OK button is clicked.
2	Specifies that Cancel button is clicked.
3	Specifies that Abort button is clicked.
4	Specifies that Retry button is clicked.
5	Specifies that Ignore button is clicked.
6	Specifies that Yes button is clicked.
7	Specifies that No button is clicked.

This will produce the result as in Fig 3.

Fig 3

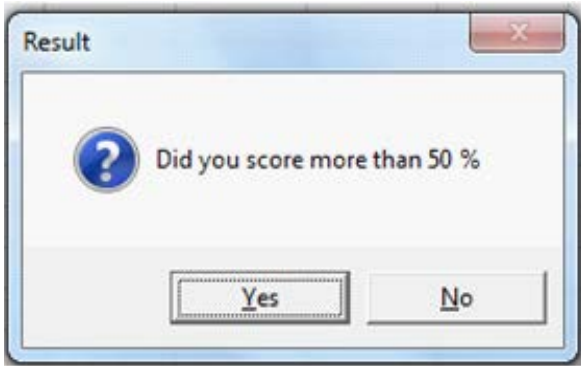


Reading the Msgbox() return values

Based on the value returned by the MsgBox(), decisions can be made.

For ex, the code mentioned here will display the message box, and when the user clicks "Yes" it will display a congratulatory message. If the user clicks "No" another message "Better Luck Next time" will appear as shown in Fig 4.

Fig 4




```
Sub test()

Dim n As Integer

n = MsgBox("Did you score more than 50 % ", vbYesNo +
vbQuestion, "Result")

If n = 6 Then

MsgBox ("Congratulations")

Else

MsgBox ("Better Luck Next Time")

 End If

End Sub
```

Input box

For accepting the input from the user the Input box is used in two ways- The Input Box Function and the Input Box Method.The InputBox method differs from the InputBox function in that it allows selective validation of the user's input, and it can be used with Microsoft Excel objects, error values, and formulas.

Note that Application.Input Box calls the Input Box method; Input Box with no object qualifier calls the InputBox function.

Input Box Function

The Input Box Function displays a dialog box for user input. It returns the information entered in the dialog box. The syntax for the InputBox function is:

InputBox(prompt[, title] [, default] [, xpos] [, ypos] [, helpfile, context])

In its simplest form , the input box function looks like:n = Inputbox("Enter your Age")

The InputBox Method

When we precede the Input Box Function with "Application" we get an InputBox Method that will allow us to specify the type of info that we can collect. Ie. Application.InputBox

Its Syntax is :Input Box(Prompt, Title, Default, Left, Top, HelpFile, HelpContextId, Type)

The Prompt, Title and Default are the same as in the InputBox Function. However, it is the last argument "Type" that allows us to specify the type of data we are going to collect. These are as shown below.

Type:=0 A formula

Type:=1 A number

Type:=2 Text (a string)

Type: = 4 A logical value (True or False)

Type: = 8 A cell reference, as a Range object

Type: = 16 An error value, such as #N/A

Type := 64 An array of values

The following is an example of an InputBox method

```
Sub test()

Dim n As Integer

n = Application.InputBox("Enter you age", "Personal Details", , , , , 1)

 'Exit sub if Cancel button used

If n > 60 Then

MsgBox "You are eligible for senior citizen's concession"

Else

MsgBox ("No concession")

End If

End Sub
```

Decision making statements in VBA

Objectives: At the end of this lesson you shall be able to
- **describe the decision making process using the "if... Then" statement**
- **describe the use of "ladder off" and "nested if" statement**
- **explain the use of the "select ....case" statements.**

Introduction

In a program a set of statements are normally executed sequentially in the order in which they appear. This happens when no decision making or repetitions are involved. But in reality, there may be a number of situations where we may have to change the order of execution of statements based on certain conditions being true or false. Some of the examples may be:

a To decide if a trainee is to be declared "Passed" or "Failed".

b To display the Grade achieved by a student.

c To accept input only of a particular data type like numbers.

d To decide if a number is prime or not.

e To decide if a string is a palindrome or not.

f To calculate pay, tax, commission etc. based on certain conditions etc.

g To repeat an action a certain number of times or till a certain limit is reached.

Decision making process can solve practical problems intelligently and provide useful output or feedback to the user. In order to control the program flow and to make decisions, we need to use the conditional operators and the logical operators together with the If control structure.

Decision making structures require that the programmer specify one or more conditions to be evaluated or tested by the program, along with a statement or statements to be executed if the condition is found to be true, and other statements to be executed if the condition is found to be false.Table 1 shows the commonly used decision making statements in VBA.

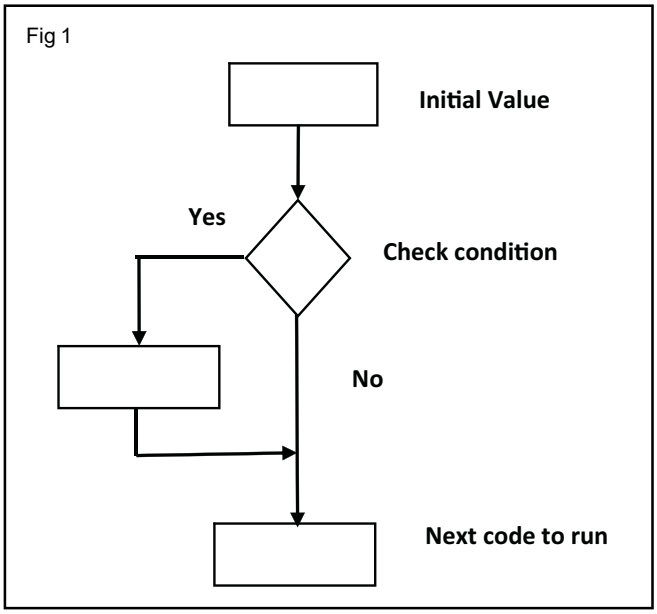
The If … Then Statement

It is the simplest form of control statement, frequently used in decision making and changing the control flow of the program execution. Syntax for if-then statement is:

```
If CONDITION Then

' code if the condition is met

End If
```

The flow chart for a typical If statement is shown in Fig 1.

Fig 1



Here condition refers to an expression which results in a Boolean type result, ie. True or False.For ex. the statement "if age <18" will test if the value of the variable "age" is less than 18 or not. If the condition evaluates to true, then the block of code inside the If statement will be executed. For example:

```
If (age < 18) Then

debug.print "Not Eligible"

End If
```

The following example tests the value of the number in the textbox and takes a decision.

```
Private Sub Button1_Click()

Dim n As Integer

'Enter the number of items sold by the agent

n = val(TextBox1.Text)

If n> 100 Then

Label1.Caption = " You are entitled for a commission of Rs. 10000"
```

End If

End Sub

The If Then…. Else Statements

When an action has to be taken if the condition returns true and another action if the condition returns false, then we use the If Then…. Else Statements.

The syntax for the If Then … Else statements is as follows

If CONDITION Then

' code if the condition is met

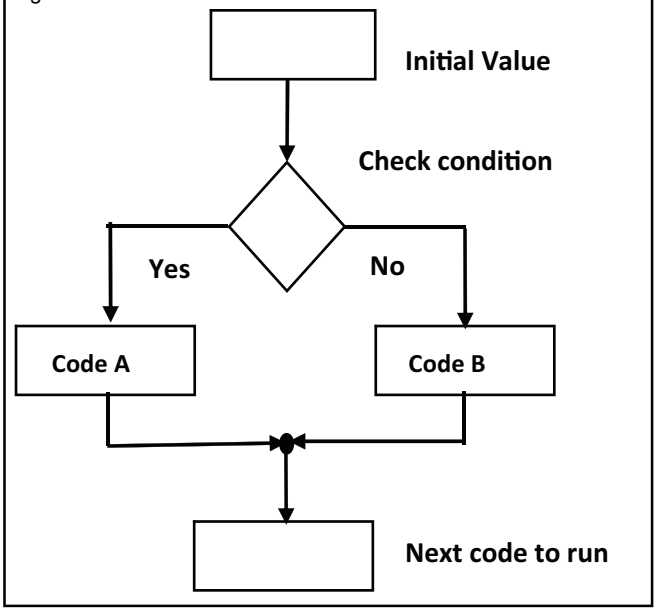
Else

' code if the condition is not met

End If

The flow chart for a typical If Then … Else structure is as shown in Fig 2.

Fig 2



The example of an If Then … Else structure is shown below. This program tests if the Taxable Income entered by the user is less than 250000 or not. If yes, a message box appears stating that the user need not pay Income Tax. Else, another message box tells the user to pay the tax.

```
Sub test()

Dim income As Long

income = Application.InputBox("Enter you Taxable income")

If income< 250000 Then

MsgBox "You need not pay any income tax"

Else

Msg Box ("You must pay income tax")

End If

End Sub
```

Using Multiple If Statements

Sometimes the condition being tested is to be evaluated not just for returning "True" or "False" based on one condition, but for multiple conditions too. In such cases the multiple If Then …. Else statements can be used. They can be used in two ways:

1 Ladder If and

2 Nested if

Ladder If statements

The ladder if statements can be used to test if a condition1, condition2 … etc is met, and decision be taken based on which condition is met. The typical syntax of a Ladder If structure is:

```
if(boolean_expression 1)

{

/* Executes when the boolean expression 1 is true */

}

else if( boolean_expression 2)

{

 /* Executes when the boolean expression 2 is true */

}

else if( boolean_expression 3)

{

 /* Executes when the boolean expression 3 is true */

}

else

{

 /* executes when the none of the above condition is true */

}
```

The following is an example of a ladder if structure.

```
Sub grades()

Dim marks As Integer

marks = InputBox("Enter you marks")
If marks >= 80 Then
MsgBox "Distinction"

ElseIf marks >= 70 Then

MsgBox "A Grade"

ElseIf marks >= 60 Then
MsgBox "B Grade"

ElseIf marks >= 40 Then

MsgBox "C Grade"

Else

MsgBox "Failed"

End If

End Sub
```

This program would display the grade based on the marks entered by the user.

Nested If statements

Sometimes it is required to evaluate one condition only if an earlier condition is met. In such cases an If Then statement can be placed inside an outer If Then statement. This type of structure is also called a Nested If structure.The syntax of a nested if structure is as follows:

```
If(Boolean_expression 1)
{
//Executes when the Boolean expression 1 is true
If(Boolean_expression 2)
{
//Executes when the Boolean expression 2 is true
}
}
```

For ex. A certain recruitment condition states that a candidate to be declared eligible must have a minimum of 5 years' experience **and also** must have scored atleast 75% marks in the exam. In such a case, the first condition to be tested is for experience >= 5 years andonly if this condition is met, the second condition is to be evaluated. If the first condition is not met, the control jumps to the statement after the End if statement. The following code is an example for the mentioned example.

```
Sub job_test()
Dim experience, marks As Integer
experience = InputBox("Enter your work experience in years")

If experience >= 5 Then
marks = InputBox("Enter you marks percentage")

If marks >= 75 Then

MsgBox (" You are eligible for the post")

Else

MsgBox (" You are NOT eligible for the post")

End If

Else

MsgBox (" You are NOT eligible for the post")

End If

End Sub
```

Using Logical operators in If Structure

The Logical operators And, Or and Not can be used in If structure and produce the same results as those produced in Nested If Structures.

For ex. the above mentioned condition can be evaluated using the And operator in the conditional statement.

```
Sub job_test()
Dim experience, marks As Integer
experience = InputBox("Enter your work experience in years")
marks = InputBox("Enter you marks percentage")
If experience >= 5 And marks >= 75 Then
MsgBox (" You are eligible for the post")
Else
MsgBox (" You are NOT eligible for the post")
End If
End Sub
```

Select...Case

Another way to implement decision making in your VBA code is to use a Select...Case statement. Select...Case statements can be used to easily evaluate the same variable multiple times and then take a particular action depending on the evaluation.

It is always a good practice to use Select Case Statement when multiple If-Else conditions are involved. As the number of If-Else conditions increases, debugging and understanding all the flow becomes a tedious job.

The syntax for a Select...Case statement is:

Select Case VARIABLE

Case VALUE1

' code to run if VARIABLE equals Value1

Case VALUE2

' code to run if VARIABLE equals Value2

Case Else

' code to run for remaining cases

End Select

For Ex. This program asks the user to type the name of the game and displays the number of players for the game.

```
Sub players()
Dim game As String
game = InputBox("enter the name of the game")
game = LCase(game)
Select Case game
Case "tennis"
Debug.Print "2 Players."
Case "cricket"
Debug.Print "11 Players."
Case "volleyball"
Debug.Print "5 Players."
Case "baseball"
Debug.Print "9 Players."
Case Else
Debug.Print "I have no idea."
End Select
End Sub
```

IIF Function

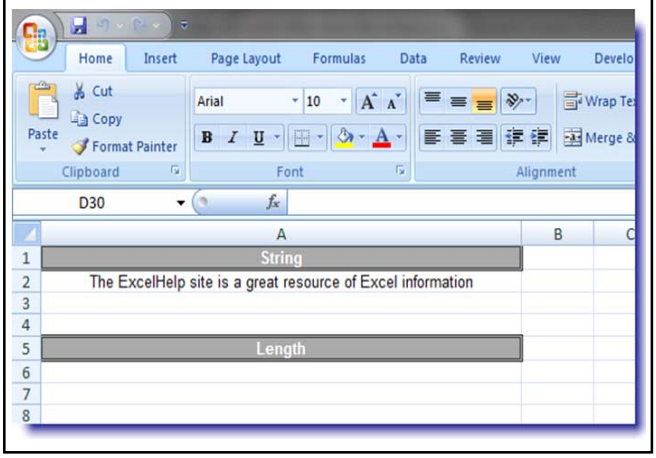
IIF function is used to evaluate an expression and perform one of two actions based on the outcome of the evaluation. For example:

IIF (Value > 10, Perform this action if Value is <= 10, Perform this action is Value is > 10)

This function is available within VBA code and also as an Excel function. Usually the IIF function is used to perform quick logical assessments and can be nested to perform more complicated evaluations. It is however important to remember that nested IF statements can become very complicated and difficult to support and maintain.

Now let's look at an example. Let's assume that we want to calculate the length of the string only if it contains the value Excel Help and Excel. (Fig 3)

Fig 3



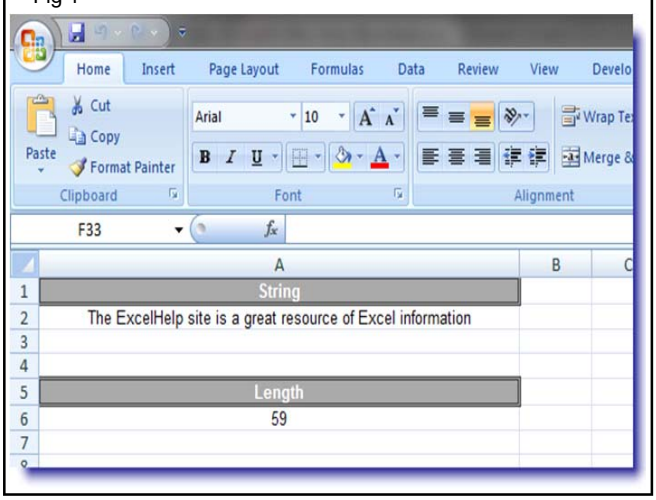
It is important to note that we could have used the IIF statement in one of our For Next loops to run through all the rows on a worksheet.

Code

```
Dim StringToProcess As String'Variable to hold the string to be processed

StringToProcess = ActiveSheet.Cells(2, 1).Value

ActiveSheet.Cells(6, 1).Value = IIf(InStr(StringToProcess, "ExcelHelp") > 0, IIf(InStr(StringToProcess, " Excel ") > 0,

Len(StringToProcess), 0), 0)
```

Output (Fig 4)

Fig 4



Looping statements in VBA

Objectives: At the end of this lesson you shall be able to
- **describe the "for" loops in VBA**
- **describe the "do" loops in VBA**
- **explain the use of the "exit" statement in VBA loops**
- **write appropriate code to perform repetitive tasks.**

Introduction

There may be many situations where you need to perform a task repeatedly / a certain number of times. In such cases the code for the task is placed inside a loop and the program iterates or repeats through the loop a certain number of times ie. till a certain condition is met. Some examples of such repetitive tasks are:

a Printing a text or number n number of times.

b Generating a sequence or series of numbers.

c Generating a table of certain calculations.

d Searching / Re arranging a set of numbers etc.

VBA provides the following types of loops to handle looping requirements (Refer Table 1)

Table 1

Loop Type	Description
for next loop	Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.
do....until loop	Repeats a statement or group of statements until a condition is met.
do….while loop	Repeats a statement or group of statements as long as the condition is true.

The For Loop

The For ... next loop sets a variable to a specified set of values, and for each value, runs the VBA code inside the loop. For Ex.

```
For n = 1 To 10

debug.print  n

Next n
```

In this example, the initial value of n is set to 1, and the loop code, ie. printing the value of n is performed.The value of n is set to the next value which is by default an increment of 1. Thus this loop is executed 10 times and would print the numbers 1 to 10. The for statement in the above code is the same as For n = 1 To 10 Step 1 since the default increment is 1

The same code will print numbers from 10 to 1 if the step is changed to a negative value as shown below.

```
For n = 10 To 1 Step -1

debug. print  n

Next n
```

Similarly, the following Ex. would add all the numbers from 1 to 10 and print the sum.

```
Dim n, sum as integer

Sum=0

For n = 1 To 10

sum=sum + n

debug. print  sum

Next n
```

The For Each Loop

The For Each loop is similar to the For ... Next loop but, instead of looping through a set of values for a variable, it loops through every object within a set of objects. The following example would print the names of all the worksheets.

```
Dim ws As Worksheet

For each ws in Worksheets

debug. print ws.name

Next ws
```

The Exit For Statement

If you need to end the For loop before the end condition is reached or met, simply use the END FOR in conjunction with the IF statement. In the example given below, we exit the for loop prematurely and before the end condition is

met. The for example given below, the loop exits when n reaches a value of 5.

```
For n = 0 To 10

debug.print n

If n=5 Then Exit For

Next n
```

The Do ….Until Loop repeats a statement or group of statements until a condition is met.

There are 2 ways a Do Until loop can be used in Excel VBA Macro code.

a Test the condition before executing the code in the loop

b Execute the code in the loop and then test for the condition.

Do Until….. Loop

In this example, the value of n is tested before going into the loop.

If the condition n=10 is not met right at the beginning itself, the code inside the loop is not executed at all. The control then jumps to the statements appearing after the Loop statement.

```
Do Until n=10

Debug.print n

n=n+1

Loop
```

Do ….. Loop Until

In this example, the code in the loop is executed at least once before testing the condition. If the condition is true, the looping stops, else the loop is executed again.

```
Do

Debug. print n

n=n+1

Loop Until n=10
```

The Do While … Loop repeats a statement or group of statements as long as the condition is true.

Like the Do until loop, a Do While loop can be also be used in two ways.

a Test the condition before executing the code in the loop

b Execute the code in the loop and then test for the condition.

Do While ….Loop

In this example, the condition ie. num<10 is checked before entering the loop. Only if the condition is met, the code in the loop is executed, otherwise it is skipped entirely. This example will print a table as shown in Fig 1.

Fig 1

number	spc(3)
1	1
2	4
3	9
4	16
5	25
6	36
7	49
8	64
9	81
10	100

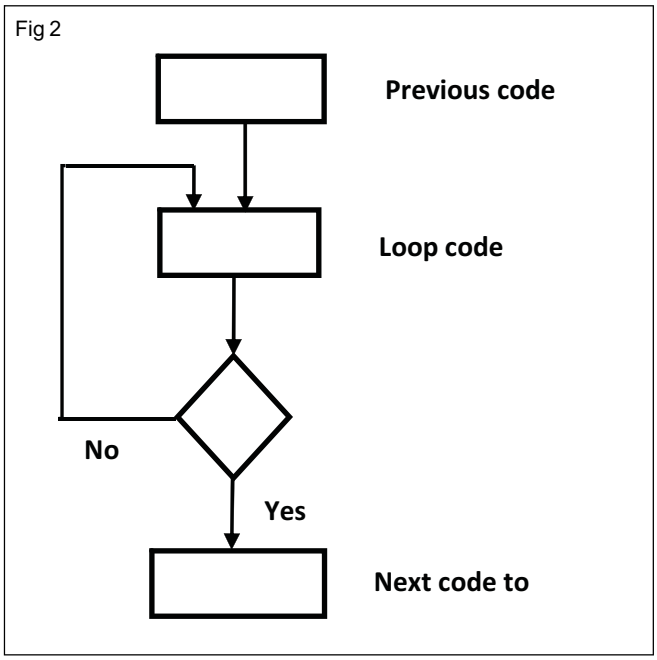
```
Dim num As Integer

Debug.Print "number"; Spc(2); "square"

Do While num < 10

num = num + 1

Debug.Print num; Spc(5); num * num

Loop
```

Do…. Loop While

In a Do…. Loop While , a set of statements in the loop are executed once, then the condition is checked. The code in the loop is executed only if the condition is met. (Refer Fig 2 for the flow chart)

In this example, the value 1 is placed in cell (1,1). The row value is incremented each time the loop code is executed. The incremented value is placed in the cell (row,1). The loop is executed as long as the row value is less than 10 after which the iterations stop. The condition checking is done after executing the loop code at least once. (Fig 2)

Fig 2




```
Dim row As Integer

row = 0

Do

row = row + 1

Cells(row, 1) = row

Loop While row < 10
```

The While ….. Wend loop

The While ….. Wend loop executes a series of statements as long as a given condition is True.

In this example the condition checking is done at the beginning of the loop. This code prints hello 5 times and then prints the value of the counter, ie. 5 at the end of the program.

```
Dim Counter

Counter = 0

While Counter < 5

Counter = Counter + 1

Debug.Print "hello"

Wend

Debug.Print Counter
```

The Exit Statement

The Exit Statement exits a procedure or block and transfers control immediately to the statement following the procedure call or the block definition. It may be in the form of Exit Do, Exit For, Exit While, Exit Select etc. depending on where it is being used. An example of an Exit statement is as follows:

```
Do While True

Count = Count + 1

Debug.Print Count

If Count = 5 Then

Debug.Print "stop at 5"

Exit Do

End If

Loop
```

In this example, the loop condition stops the loop when count=5.

Arrays in VBA

Objectives: At the end of this lesson you shall be able to
- **describe and declare an array in VBA**
- **differentiate between static and dynamic arrays**
- **declare, populate and read a multidimensional array**
- **describe the redim and preserve statements in VBA.**

Introduction

An Array is a group of variables of the same data type and with the same name. If we have a list of items which are of similar type to deal with, we need to declare an array of variables instead of using a variable for each item. For example, if we need to enter ten names, instead of declaring ten different variables for each name, we need to declare only one array holding all the names. The individual element or item in the array is identified by its index or subscript.

When arrays are used, data is stored in an organized way. Apart from this working with the data is easy and faster when iterations are done using the Loop statements like For… Next etc. on the Arrays. The following example declares an array variable to hold ten students in a school.

Dim students(10) As Integer

The array "students" in the preceding example contains ten elements. The indices of the elements range from 0 through 9 by default. The variables in the Array are now identified as students(0), students(1) etc.indicating the first element and second element etc.respectively.

Types of Arrays :

1 Static Arrays

2 Dynamic Arrays

Static array

A static array is an array that is sized in the Dim statement that declares the array. E.g.,

Dim Students(10) as String

Dim StaticArray(1 To 10) As Long

You cannot change the size or data type of a static array. When you erase a static array, no memory is freed. Erase simply sets all the elements to their default value (0, vbNullString, Empty, or Nothing, depending on the data type of the array).

Declaring an Array

You declare an array variable using the Dim statement.

Dim StudentName(3) As String

Arrays are also declared in another method where the type or the variable name with one or more pairs of parentheses is added to indicate that it will hold an array. After you declare the array, you can define its size by using the ReDim Statement.

The following example declares a one-dimensional array variable and also specifies the dimensions of the array by using the ReDim Statement.

Dim arr As Integer()

ReDim arr(10)

The following example declares a multidimensional array variable by placing commas inside the parentheses to separate the dimensions.

Dim arrayName (num1, num2) as datatype

To declare a jagged array variable, add a pair of parentheses after the variable name for each level of nested array.

Dim arr()()() As integer

In VBA arrays you can specify any value for the lower and upper bounds of the array. Element 0 need not be the first element in the array. For example, the following is perfectly legal code (as long as the lower bound is less than or equal to the upper bound -- an error is generated if the lower bound is greater the upper bound).

If you don't explicitly declare the lower bound of an array, the lower bound will be assumed to be either 0 or 1, depending on value of the Option Base statement, if present. If Option Base is not present in the module, 0 is assumed. For example, the code

Dim Arr(10) As Long

declares an array of either 10 or 11 elements. The declaration does not specify the number of elements in the array. Instead, it specifies the upper bound of the array. If your module does not contain an "Option Base" statement, the lower bound is assumed to be zero, and the declaration above is the same as : Dim Arr(0 To 10) As Long

If you have an Option Base statement of 0 or 1, the lower bound of the array is set to that value.

Thus, the code : Dim Arr(10) As Long is the equivalent of either Dim Arr(0 To 10) As Long or Dim Arr(1 To 10) As Long, depending on the value of the Option Base.

It is a good programming practice to specify both the lower and upper bounds of the array to avoid bugs when copying and pasting code between modules or elsewhere.

Storing values in an array

Arrays can be populated in the following ways

1 Marks(0)=55
  Marks(1)=67
  Marks(2)=55
  Marks(3)=67
  Marks(4)=74
2 Dim marks As Integer() = {55, 67, 87, 48, 90, 74}
3 Dim marks = New Integer() {1, 2, 4, 8}
4 Dim doubles = {1.5, 2, 9.9, 18}

You can explicitly specify the type of the elements in an array that's created by using an array literal. In this case, the values in the array literal must widen to the type of the elements of the array. The following code example creates an array of type Double from a list of integers: Dim marks As Double() = {55, 67, 87, 48, 90, 74}

Iterating through an Array

Loop statements like for… next, Do …while etc. can be used with arrays to retrieve their values.An example of such a code is shown below.

```
Sub array_test()
Dim arr(5) As Integer
Dim n As Integer
arr(0) = 89
arr(1) = 56
arr(2) = 78
arr(3) = 45
arr(4) = 99
For n = LBound(arr) To UBound(arr) - 1
Debug.Print arr(n)
Next
End Sub
```

Here LBound() and UBound() functions return the Lower and Upper Bounds of the Array.

Multi dimensional Arrays

Multi dimensional arrays have more than one row or one column.

For ex.Dim MyArray(5, 4) As Integer

Dim MyArray(1 To 5, 1 To 6) As Integer

In the following ex. we will define an array with 3 elements each in two rows

```
Sub array_test()
Sub array_test()
Dim m, n As Integer
Dim arr(2, 4) As String
arr(0, 0) = "printer"
arr(0, 1) = "scanner"
arr(0, 2) = "mouse"
arr(0, 3) = "monitor"
arr(1, 0) = "usb"
arr(1, 1) = "ps2"
arr(1, 2) = "firewire"
arr(1, 3) = "serial"
For m = 0 To 2
   For n = 0 To 3
  Debug.Print arr(m, n); Spc(2);
   Next n
  Debug.Print
Next m
End Sub
```

Dynamic Arrays

A dynamic array is an array that is not sized in the Dim statement. Instead, it is sized with the ReDim statement. Dynamic array variables are useful when we don't know in advance how many elements need to be stored in the array or when we need to change the array dimensions at a later stage.

E.g. : Dim DynamicArray() As Long

ReDim DynamicArray(1 To 10)

If an array is sized with the ReDim statement, the array is said to be allocated(either static array or a dynamic array).Static arrays are always allocated and never empty.You can change the size of a dynamic array, but not the data type. When you Erase a dynamic array, the memory allocated to the array is released. You must ReDim the array in order to use it after it has been Erased.

If a dynamic array has not yet been sized with the ReDim statement, or has been deallocated with the Erase statement, the array is said to be empty or unallocated. Static arrays are never unallocated or empty.

ReDim Statement:

You may declare a dynamic variable with empty parentheses ie. leave the index dimensions blank. You can thereafter size or resize the dynamic array that has already been declared, by using the ReDim statement. To resize an array, it is necessary to provide the upper bound, while the lower bound is optional. If you do not mention the lower bound, it is determined by the Option Base setting for the module, which by default is 0. You can specify Option Base 1 in the Declarations section of the module and then index will start from 1. This will mean that the respective index values of an array with 3 elements will be 1, 2 and 3. Not entering Option Base 1 will mean index values of 0, 1 and 2.

The following example declares an array called A1 as a dynamic array. The array's size is not set and then it is resized to 3 elements (by specifying Option Base 1)

```
Sub arr_test()

'declare a dynamic array

Dim A() As String

ReDim A(3) As String

A(1) = "COPA"

A(2) = "DTPO"

A(3) = "MASE"

debug.print  A(1)  & " , " & A(2) & " , " & A(3)

End Sub
```

When you use the ReDim keyword, you erase any existing data currently stored in the array.

For ex. add another element to the array mentioned in the example above using the redim statement as follows and assign a value to it.

```
ReDim A(4) As String

A(4) = "CHNM"
```

Now when the array values are displayed again, the earlier values will all be blank, since they are erased by the redim statement. The example below shows this:

```
Sub arr_test()

'declare a dynamic array

Dim A() As String

ReDim A(3) As String

A(1) = "COPA"

A(2) = "DTPO"

A(3) = "MASE"

ReDim A(4) As String

A(4) = "CHNM"

Debug.Print A(1) & " , " & A(2) & " , " & A(3) & "," & A(4)

End Sub
```

The result of this program will be , , ,CHNM

To resize the array without losing the existing data, you should use " Preserve " along with Redim. For ex. ReDim Preserve A(4) As String.

String manipulation in VBA

Objectives: At the end of this lesson you shall be able to
- **describe the string concatenation functions in VBA**
- **describe the string conversion functions in VBA**
- **describe the string extraction functions in VBA**
- **describe the string formatting functions in VBA.**

Introduction

A string, in VBA, is a type of data variable which can consist of text, numerical values, date and time and alphanumeric characters. Strings are a frequently used to store all kinds of data and are important part of VBA programs. To declare a variable for it, you can use either String or the Variant data types.

String Manipulation

VBA has a robust set of functions for string handling. The following are some examples of where you might use string functions:

- Checking to see whether a string is contained another string
- Parsing out a portion of a string
- Replacing parts of a string with another value
- Finding the length of the string etc.

String Concatenation

String concatenation or joining two or more strings can be done by the "+" Operator

Example : Dim A, B, C As String

A = "www"

B = " and"

C = " the Internet"

Debug.Print A + B + C

This will print "www and the Internet"

If it is required to join two or more different data types variable, then the "&" operator can be used.

Example : Dim person As String, pay As Integer

person = "Jaya"

pay = 25000

Debug.Print "The payment for "& person & " is " & pay

This will print: The payment for Jaya is 25000

Len() function

Returns an integer containing either the number of characters in a string or the nominal number of bytes required to store a variable.

Syntax : Len (String)

Example : 1

Dim str As String

str = "Computer operator and programming assistant"

Debug.Print "The length of the string is "& Len(str)

This will print : The length of the string is 43

Example : 2

Dim p, q As Integer, r As Double, dob As Date

p = Sqr(25)

q = 3333

r = 45.6789

dob = #1/1/1990#

Debug.Print "Size of p Is : " & Len(p)

Debug.Print "Size of q Is : " & Len(q)

Debug.Print "Size of r Is : " & Len(r)

Debug.Print "Size of dob Is : " & Len(dob)

Thiswill print : Size of p Is : 1

Size of q Is : 2

Size of r Is : 8

Size of dob Is : 8

Left()

Returns a string containing a specified number of characters from the left side of a string.

Syntax: Left(String ,Int)

Right()

Returns a string containing a specified number of characters from the right side of a string.

Syntax: Right(String, Int)

Mid()

Returns a string that contains characters from a specified string.

Syntax:Mid(String, Int, Int)

Returns a string that contains all the characters starting from a specified position in a string

Syntax: Mid(String, Int, Int)

Returns a string that contains a specified number of characters starting from a specified position in a string.Examples are :

1 Dim s AsString
s =" Indiana Jones"
debug.print Left(s)
This will print : India

2 Dim s AsString
s =" FUNDAMENTALLY"
debug.print right(s, 5)
This will print : TALLY

3 Dim s AsString
s =" wholehearted"
debug.print mid(s, 6)
This will print : hearted

4 Dim s AsString
s =" wholehearted"
debug.print mid(s, 6,4)
This will print : hear

Ltrim()

Returns a string containing a copy of a specified string with no leading spaces (LTrim)

Syntax :LTrim(String)

RTrim()

Returns a string containing a copy of a specified string with no trailing spaces.

Syntax :RTrim(String)

Trim()

Returns a string containing a copy of a specified string with no leading or trailing spaces.

Syntax : Trim(String)

Examples: Dim A as String

A = " Adjustment "

Debug.Print "For everyone" &LTrim(A) & "is a must"

Debug.Print "For everyone"; RTrim(A) & "is a must"

Debug.Print "For everyone"; Trim(A) & "is a must"

This will print : For everyoneAdjustment is a must

For everyone Adjustmentis a must

For everyone Adjustmentis a must

Instr()

Returns an integer specifying the start position of the first occurrence of one string within another.

Syntax: InStr([start,]string1, string2[, compare])

Example:

A = "hairdresser"

B = "dress"

Debug.Print "The second string starts at position no. " &InStr(1, A, B) "

This will print : The second string starts at position no. 5

Replacing Strings

The Replace() function replaces a sequence of characters in a string with another set of characters.

Replace(source_string, find_string, replacement_string).

Eample: Debug.Print Replace("majordrawback", "drawback", "advantage")

Debug.Print Replace("majority", "aj", "in", 1)

Debug.Print Replace("think and think", "i", "a", 1, 1)

This will print : major advantage

minority

thank and think

Val()

The VAL() function accepts a string as input and returns the numbers found in that string.The VAL function will stop reading the string once it encounters the first non-numeric character. This does not include spaces.

Syntax: Val(String)

Example: Dim s1, s2, s3 As String

s1 = "6 feet 1 inch is his height"

s2 = "5 - 6 kms is the distance to my office from here"

s3 = "011 22222222 is my telephone number"

Debug.Print Val(s1)

Debug.Print Val(s2)

Debug.Print Val(s3)

This will print : 6

5

1122222222

The String Conversion Functions

LCase()

Returns a string or character converted to lowercase.

Syntax :LCase(String)

UCase()

Returns a string or character converted to uppercase.

Syntax :UCase(String)

Example:

Dim A, B as String

A="IF YOU FEAR YOU WILL BECOME WEAK"

B = "be a strong person"

Debug.PrintLcase(A)

Debug.PrintUCase(B)

This will print: if you fear you will become weak

BE A STRONG PERSON

Str()

The Str() function converts a number to a string.

CStr()

The CStr() function is used to convert any type of value to a string.

Syntax: Str(number as variant)

Example:

```
1  Dim Number As Double
   Number = 1450.5
   Debug.Print "The string is "&str(Number)
   This will print : The string is 1450.5
2  Dim Date_of_birthAs Date
   Date_of_birth = #1/1/1990#
   Debug.Print CStr(Date_of_birth)
   This will print : 01/01/1990
```

Asc()

The Asc() function returns an Integer value representing the ASCII code corresponding to a character or the first character in a string

Syntax :Asc(String)

Example :Asc("A") will return 65

Chr()

The Chr() Function returns the character associated with the specified ASCII code.

Syntax :Chr(Integer)

Example :Chr(68) will return the character "D"

Reversing a String

StrReverse(String)

StrReverse() returns a string in which the character order of a specified string is reversed.

Example: Dim A As String

A = "desserts"

Debug.Print StrReverse(A)

This will print : stressed

Format() function

The format() function returns a Variant (String) containing an expression formatted according to instructions contained in a format expression. It can be used to return formatted dates as well as formatted strings.

Syntax(for FormattingStrings) : Format(String, Format)

User-Defined String Formats (Format Function)

You can use any of the following characters to create a format expression for strings:

Example :Dim x As String

Character	Description
@	Character placeholder. Display a character or a space. If the string has a character in the position where the at symbol (@) appears in the format string, display it; otherwise, display a space in that position. Placeholders are filled from right to left unless there is an exclamation point character (!) in the format string.
&	Character placeholder. Display a character or nothing. If the string has a character in the position where the ampersand (&) appears, display it; otherwise, display nothing. Placeholders are filled from right to left unless there is an exclamation point character (!) in the format string.
<	Force lowercase. Display all characters in lowercase format.
>	Force uppercase. Display all characters in uppercase format.
!	Force left to right fill of placeholders. The default is to fill placeholders from right to left.

x = "change case"

Debug.Print Format(x, ">")

This will print "CHANGE CASE"

Built in Functions in VBA

Objectives: At the end of this lesson you shall be able to
- **describe the math functions in VBA**
- **describe the logical functions in VBA**
- **describe the date/time functions in VBA**
- **describe the conversion functions in VBA.**

Introduction

VBA has a rich collection of built in functions that perform a variety of tasks and calculations for you. There are functions to convert data types, perform calculations on dates, perform simple to complex mathematics, make financial calculations, manage text strings, format values, and retrieve data from tables, among others. Using the VBA Built in Functions will help coding much easier for the user. We have already used many built in functions in our earlier lessons like the msgbox function and many string manipulation functions just to name a few.

MS Excel: VBA Functions (VBA Formulae) - Category wise

The commonly used VBA functions in Excel, sorted by Category are shown here.

String Functions: The string functions were already discussed in the related theory for Ex. 2.2.09

Math Functions

Table 1 : Lists some of the common Built in Functions in the Mathematical category.

Table 1

Function Name	Description
Abs	Returns the absolute value of a number.
Cos	Returns the cosine of the specified angle.
Cosh	Returns the hyperbolic cosine of the specified angle.
Exp	Returns e (the base of natural logarithms) raised to the specified power.
Fix	Returns the integer portion of a number.
Format	Takes a numeric expression and returns it as a formatted string.
Int	Returns the integer portion of a number.
Log	Returns the natural (base e) logarithm of a specified number or the logarithm of a specified number in a specified base.
Rnd	Generates a random number (integer value)
Round	Returns a Decimal or Double value rounded to the nearest integral value or to a specified number of fractional digits.
Sign	Returns an Integer value indicating the sign of a number.
Sin	Returns the sine of the specified angle.
Sqr	Returns the square root of a specified number.
Tan	Returns the tangent of the specified angle.
Val	Accepts a string as input and returns the numbers found in that string.

1 Dim a, b as integer

a=81

debug.print sqr(a)

This will display the square root of 81 ie. 9

Logical Functions

Table 2. lists some of the common Built in Functions in the Logical category.

Table 2

Function Name	Description
ISDATE	Returns TRUE if the expression is a valid date. Otherwise, it returns FALSE.
ISERROR	Checks for error values.
ISNULL	Returns TRUE if the expression is a null value. Otherwise, it returns FALSE.
ISNUMERIC	Returns TRUE if the expression is a valid number. Otherwise, it returns FALSE.

Examples:

1 Sub Button1_Click()

N = TextBox1.Text

If IsNumeric(N) = True Then

MsgBox "correct"

Else

MsgBox "Insert only numbers"

End If

This checks if the data entered in the textbox is a number or not.

2 Sub Button1_Click()

N = TextBox1.Text

If IsDate(N) = True Then

MsgBox "correct"

Else

MsgBox "Insert only dates"

End If

End Sub

This checks if the data entered in the textbox is a valid date or not.

Date / Time Functions

Table 3. lists some of the common Built in Functions in the Date / Time category.

Table 3

Function	Return Value
DATE	Returns the current system date.
DATEADD	Returns a date after which a certain time/date interval has been added.
DATEDIFF	Returns the difference between two date values, based on the interval specified.
DATEPART	Returns a specified part of a given date.
DATESERIAL	Returns a date given a year, month, and day value.
DATEVALUE	Returns the serial number of a date.
DAY	Returns the day of the month (a number from 1 to 31) given a date value.
FORMAT Dates	Takes a date expression and returns it as a formatted string.
HOUR	Returns the hour of a time value (from 0 to 23).
MINUTE	Returns the minute of a time value (from 0 to 59).
MONTH	Returns the month (a number from 1 to 12) given a date value.
MONTHNAME	Returns a string representing the month given a number from 1 to 12.
NOW	Returns the current system date and time.
TIMESERIAL	Returns a time given an hour, minute, and second value.
TIMEVALUE	Returns the serial number of a time.
WEEKDAY	Returns a number representing the day of the week, given a date value.
WEEKDAYNAME	Returns a string representing the day of the week given a number from 1 to 7.
YEAR	Returns the year portion of the date argument.

Examples

1 DateDiff() Function

Syntax for the DateDiff function is :

DateDiff (interval, date1, date2, [firstdayofweek], [firstweekofyear])

Parameters or Arguments

Interval is the interval of time to use to calculate the difference between date1 and date2. Below is a list of valid interval values as in Table 4

Table 4

Interval	Explanation
yyyy	Year
q	Quarter
m	Month
y	Day of year
d	Day
w	Weekday
ww	Week
h	Hour
n	Minute
s	Second

Date1 and Date2 are the two dates to calculate the difference between.

first day of week is optional. It is a constant that specifies the first day of the week. If this parameter is omitted, Excel assumes that Sunday is the first day of the week.

first week of year is optional. It is a constant that specifies the first week of the year. If this parameter is omitted, Excel assumes that the week containing Jan 1st is the first week of the year.

```
Sub test()

Debug.PrintDateDiff("yyyy", "1/12/1999", "31/1/2000")

Debug.PrintDateDiff("q", "1/12/1999", "31/1/2000")

Debug.PrintDateDiff("m", "1/12/1999", "31/1/2000")

End Sub
```

The result will be

1

4

12

2 Format Date

Syntax

The syntax for the Microsoft Excel FORMAT function is:

Format (expression, [format, [firstdayofweek, [firstweekofyear]]])

Parameters or Arguments

Expression is the value to format.

Format is optional. It is the format to apply to the expression. You can either define your own format or use one of the named formats that Excel has predefined such as shown in Table 5.

Table 5

Format	Explanation
General Date	Displays date based on your system settings
Long Date	Displays date based on your system's long date setting
Medium Date	Displays date based on your system's medium date setting
Short Date	Displays date based on your system's short date setting
Long Time	Displays time based on your system's long time setting
Medium Time	Displays time based on your system's medium time setting
Short Time	Displays time based on your system's short time setting

First day of week is optional. It is a value that specifies the first day of the week. If this parameter is omitted, the FORMAT function assumes that Sunday is the first day of the week. This parameter can be one of the following values as shown in Table 6.

First week of year is optional. It is a value that specifies the first week of the year. If this parameter is omitted, the FORMAT function assumes that the week that contains January 1 is the first week of the year. This parameter can be one of the following values as shown in Table 7.

```
Sub test()

Debug.Print Format(#1/1/1990#, "Short Date")

Debug.Print Format(#1/1/1990#, "Long Date")

Debug.Print Format(#1/1/1990#, "yyyy/mm/dd")

End Sub
```

Table 6

Constant	Value	Explanation
vbUseSystem	0	Uses the NLS API setting
VbSunday	1	Sunday (default, if parameter is omitted)
vbMonday	2	Monday
vbTuesday	3	Tuesday
vbWednesday	4	Wednesday
vbThursday	5	Thursday
vbFriday	6	Friday
vbSaturday	7	Saturday

The result will be

1/1/1990

Table 7

Constant	Value	Explanation
vbUseSystem	0	Uses the NLS API setting
vbFirstJan1	1	The week that contains January 1
vbFirstFourDays	2	The first week that has at least 4 days in the year
vbFirstFullWeek	3	The first full week of the year

Monday, January 01, 1990

1990/01/01

Data Type Conversion Functions

Table 8. below lists some of the common Built in Functions in the Data Type Conversion category.

Table 8

Function	Return Type	Range for expression argument
CBool	Boolean	Any valid string or numeric expression.
CByte	Byte	0 to 255.
CCur	Currency	-922,337,203,685,477.5808 to 922,337,203,685,477.5807.
CDate	Date	Any valid date expression.
CDbl	Double	-1.79769313486231E308 to -4.94065645841247E-324 for negative values; 4.94065645841247E- 324 to 1.79769313486232E308 for positive values.
CDec	Decimal	+/-79,228,162,514,264,337,593,543,950,335 for zero-scaled numbers, that is, numbers with no decimal places. For numbers with 28 decimal places, the range is +/-7.9228162514264337593543950335. The smallest possible non-zero number is 0.0000000000000000000000000001.
CInt	Integer	-32,768 to 32,767; fractions are rounded.
CLng	Long	-2,147,483,648 to 2,147,483,647; fractions are rounded.
CSng	Single	-3.402823E38 to -1.401298E-45 for negative values; 1.401298E-45 to 3.402823E38 for positive values.
CStr	String	Returns for CStr depend on the expression argument.
CVar	Variant	Same range as Double for numerics. Same range as String for non-numerics.

Example

CDate function

```
Sub test()
Dim INum As Long
Dim a As String
a = 12345
Debug.PrintCDate(a)
b = "January 1, 1990"
Debug.PrintCDate(b)
c = "1:23:45 PM"
Debug.PrintCDate(c)
End Sub
```

This will display

10/18/1933

1/1/1990

1:23:45 PM

User defined functions in VBA

Objectives: At the end of this lesson you shall be able to
- **create user defined functions**
- **describe passing values to functions byval and byref**
- **describe using arrays with functions**
- **describe the scope of variables**
- **describe the access specifiers public and private.**

Introduction

In Excel Visual Basic too, like in most programming languages, a set of commands to perform a specific task is placed into a procedure, which can be a function or a subroutine. The main difference between a VBA function and a VBA subroutine is that a function (generally) returns a result, whereas a subroutine does not.

Therefore, if you wish to perform a task that returns a result (ex. summing of a group of numbers), you will generally use a function, but if you just need a set of actions to be carried out (ex. formatting a set of cells), you might choose to use a subroutine.

User Defined Functions

One of the most power features of Excel VBA is that you can create your own functions or UDFs. A UDF (User Defined Function) is simply a function that you create yourself with VBA for your own defined tasks. UDFs are often called "Custom Functions". A UDF can remain in a code module attached to a workbook, in which case it will always be available when that workbook is open. Alternatively you can create your own add-in containing one or more functions that you can install into Excel. Here the user-defined functions can be entered into any cell or on the formula bar of the spreadsheet just like entering the built-in formulas of the MS Excel spreadsheet.

Custom functions, like macros, use the Visual Basic for Applications (VBA) programming language. They differ from macros in two significant ways. First, they use function procedures instead of sub procedures. They start with a Function statement instead of a Sub statement and end with End Function instead of End Sub. Second, they perform calculations instead of taking actions. Certain kinds of statements (such as statements that select and format ranges) are generally excluded from custom functions.

A simple function may look like this:

```
Function area()

Dim l, b

l = 10

b = 20

Debug.Print "area Is " & l * b

End Function
```

When executed from the immediate window this function displays the area.

Alternately this function can be called by another subroutine, for ex.

```
Sub test_fn()

Call area

End Sub
```

Returning a value from the procedures

In the example given below, the area() function calculates l*b.

The subroutine that calls this function is returned this value.

```
Sub test_fn()

Debug.Print "The function has returned the value " & area

End Sub

Function area()

Dim l, b, A

l = 10

b = 20

area = l * b

End Function
```

The result will be:The function has returned the value 200

Passing Arguments to functions

We can pass the arguments in two different ways:

1 By Value (ByVal): We pass the copy of the actual value to the arguments

2 By Reference (ByRef): We pass the reference to the arguments

By Ref is the default method of passing argument type in VBA. This means, if you are not specifying any type of the argument it will consider it as ByRef type. However, it is always a good practice to specify the ByRef even if it is not mandatory.

The following example shows the method of passing variables to a function byVal.

```
Sub test_fn()

Dim a, b As Integer

a = 4

b = multiply(a)

Debug.Print "a is " & a

Debug.Print "The function has returned the value " & b

End Sub

Function multiply(ByVal a As Integer)

a = a * 10

multiply = a

End Function
```

The result of this program will be :

```
a is 4

The function has returned the value 40

a is 4
```

This means that the value of the variable that was passed is not disturbed by the function.

The following example shows the method of passing variables to a function byRef.

```
Sub Test()

Dim A As Integer

A = 10

Debug.Print "The function has returned the value " & Modify(A)

Debug.Print "A is now  " & A

End Sub

Function Modify(ByRef A As Integer)

A = A * 2

  Modify = A

End Function
```

The result will be:

```
The function has returned the value 20

A is now 20
```

Calling a User Defined Function from Worksheet:

You call the user defined functions as similar to the built-in excel function. To do this type the arguments in the cells and type the name of the function as is done with normal functions in Excel.

Passing Arrays to User Defined functions

A Function can accept an array as an input parameter. Arrays are always passed by reference (ByRef). You will receive a compiler error if you attempt to pass an array ByVal. This means that any modification that the called procedure does to the array parameter is done on the actual array declared in the calling procedure.

(If you need to pass an array ByVal then you would need to use the Variant data type.)

An example of passing an array to a function is as follows:

```
Sub test()

Dim arr(1 To 10) As Integer

Dim i As Integer

'populates the array with the values 1 to 10

For i = 1 To 10

arr(i) = i

Next i

'call the function example 1 with arrIntegers as an input parameter

Call fn1(arr)

For i = 1 To 10

Debug.Print arr(i); Spc(2);

Next i

End Sub
```

```
'prints the values in arrIntegers to column A
Sub fn1(ByRef arr() As Integer)
Dim i As Integer
For i = LBound(arr) To UBound(arr)
  arr (i) = arr (i) * 2
  Cells (i,1) = arr (i)
Next i
End Sub
```

Scope of variables

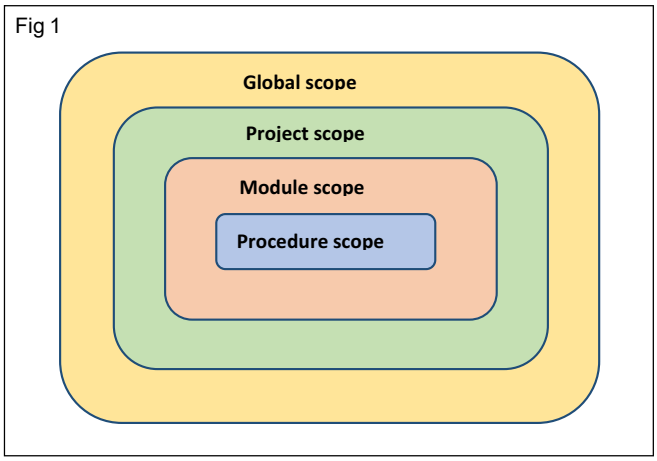
The term Scope is used to describe how a variable may be accessed. Depending on where and how a variable is declared, it may be accessible only to a single procedure, to all procedures within a module, and so on up the hierarchy of a project or group of related projects. The term visibilty is also is sometimes used to describe scope.

There are four levels of Scope:

- Procedure-Level Scope
- Module-Level Scope
- Project-Level Scope
- Global-Level Scope

Fig 1 shows the various scopes and their levels.

Fig 1



Procedure (local) scope

A local variable with procedure scope is recognized only within the procedure in which it is declared. A local variable can be declared with a Dim or Static statement.

When a local variable is declared with the Dim statement, the variable remains in existence only as long as the procedure in which it is declared is running. Usually, when the procedure is finished running, the values of the procedure's local variables are not preserved, and the memory allocated to those variables is released. The next time the procedure is executed, all of its local variables are reinitialized.

For Example the following subroutine has been created in Module1 code.

```
Sub disp()
Dim s As string
s="hello"
MsgBox s
End Sub
```

Run the subroutine "disp" in Module1 and it will display the message "Hello" in the message box.

Now the following subroutine has been created in Sheet1 code to call the disp() subroutine from Module1.

```
Sub Button1_Click()
disp
End Sub
```

This will generate an error since the subroutine disp() and the variable s are local to Module1 and cannot be accessed from elsewhere.

Static:

A local variable declared with the Static statement remains in existence the entire time Visual Basic is running. The variable is reset when any of the following occur:

- The macro generates an untrapped run-time error.
- Visual Basic is halted.
- You quit Microsoft Excel.
- You change the module.

For example, in the FindTotal example, the Accumulate variable retains its value every time it is executed. The first time the module is run, if you enter the number 2, the message box will display the value "2." The next time the module is run, if the value 3 is entered, the message box will display the running total value to be 5.

```
Sub FindTotal()
Static Total
Dim n as integer
n =InputBox("Enter a number: ")
Total = Total + n
MsgBox "The total is " &n
End Sub
```

Module scope

A variable that is recognized among all of the procedures on a module sheet is called a "module-level" variable. A module-level variable is available to all of the procedures in that module, but it is not available to procedures in other modules. A module-level variable remains in existence while Visual Basic is running until the module in which it is declared is edited. Module-level variables can be declared with a Dim or Private statement at the top of the module above the first procedure definition.

At the module level, there is no difference between Dim and Private. Note that module-level variables cannot be declared within a procedure.

Note If you use Private instead of Dim for module-level variables, your code may be easier to read (that is, if you use Dim for local variables only, and Private for module-level variables, the scope of a particular variable will be more clear).

In the following example, two variables, A and B, are declared at the module level. These two variables are available to any of the procedures on the module sheet. The third variable, C, which is declared in the Example3 macro, is a local variable and is only available to that procedure.

Note that in Test4, when the macro tries to use the variable C, the message box is empty. The message box is empty because C is a local variable and is not available to Test4, whereas variables A and B are.

```
Dim A As Integer       ' Module-level variable.

Private B As Integer   ' Module-level variable.

Sub Test1()

A = 10

B = A * 10

End Sub

Sub Test2()

MsgBox "The value of A is " & A

MsgBox "The value of B is " & B

End Sub

Sub Test3()

Dim C As Integer    ' Local variable.

C = A + B

MsgBox "The value of C is " & C

End Sub

Sub Test4()

MsgBox A

MsgBox B

MsgBox C

 ' The message box is blank since C is a local variable.

End Sub
```

Project Scope

Project scope variables are those declared using the Public keyword. These variables are accessible from any procedure in any module in the project. In Excel, a Project is all of the code modules, userforms, class modules, and object modules (e.g. ThisWorkbook and Sheet1) that are contained within a workbook.

In order to make a variable accessible from anywhere in the project, you must use the Public keyword in the declaration of the variable. However, this makes the variable accessible to any other project that reference the project containing the variable. If you want a variable to be accessible from anywhere within the project, but not accessible from another project, you need to use Option Private Module as the first line in the module (above and outside of any variable declaration or procedure). This option makes everything in the module accessible only from within the project. The project variables that should not be accessible to other projects should be declared in a module that has the Option Private Module directive. Variables that should be accessible to other project should be declared in a different module that does not use the Option Private Module directive. In both cases, however, you need to use the Public keyword.

Global Scope

Global scope variables are those that are accessible from anywhere in the project that declares them as well as any other project that references the first project. To declare a variable with global scope, you need to declare it using the Public keyword in a module that does not use the Option Private Module directive. In order to access variables in another project, you can simply use the variable's name. If, however, it is possible that the calling project also has a variable by the same name, you need to prefix the variable name with the project name. For example, if Project1 declares a global variable named x, and Project2 references Project1, code that is in Project2 can access x with either of the following lines of code:

```
x = 78

Project1.x = 78
```

If both Project1 and Project2 have variables with at least project scope, you need to include the project name with the variable. For clarity and maintainability, you should always include the project name when accessing a variable that is declared in another project. Even if this is not necessary, it makes the code more readable and maintainable.

There is no way to give some variables project, but not global, scope and give others in the same module global scope. Project versus global scope is handled only at the module level, not at the variable level.

The Access Specifiers

One of the techniques in object-oriented programming is encapsulation. It concerns the hiding of data in a class and making them available only through its methods. Most programming languages implementing OOPS allow you to control access to classes, methods, and fields via so-called access modifiers. The access to classes, constructors, methods and fields are regulated using access modifiers i.e. a class can control what information or data can be accessible by other classes. The VBA access specifiers are:

1 Private

2 Public

A Public procedure is accessible to all code inside the module and all code outside the module, essentially making it global. A VBA Private Sub can only be called from anywhere in the Module in which it resides. A Public Sub in the objects, ThisWorkbook, ThisDocument, Sheet1, etc. cannot be called from anywhere in a project. However, if you declare a Module level variable with the Public Keyword it can be used anywhere in the project and retains its value.

If you exclude the key word private in your declaration then by default the procedure is public. So Sub MySub() and Public Sub MySub() are exactly the same thing.

Public [variable] means that the variable can be accessed or used by subroutines in outside modules. These variables must be declared outside of a subroutine (usually at the very top of your module). You can use this type of variable when you have one subroutine generating a value and you want to pass that value on to another subroutine stored in a separate module.

A Private procedure is only available to the current module. It cannot be accessed from any other modules, or from the Excel workbook.Private Sub sets the scope so that subroutines from outside modules cannot call that particular subroutine. This means that a sub in Module 1 could not use the Call method to initiate a Private Sub in Module 2.

Private [variable] means that the variable cannot be accessed or used by subroutines in other modules. In order to be used, these variables must be declared outside of a subroutine (usually at the very top of your module). You can use this type of variable when you have one subroutine generating a value and you want to pass that value on to another subroutine in the same module.

Dim[variable] is used to state the scope inside of a subroutine (you cannot use Private in its place). Dim can be used either inside a subroutine or outside a subroutine (using it outside a subroutine would be the same as using Private).

Example of Public, Private Variables and Procedures.

Module 1 code

```
Dim x As Integer ' This is a Private Variable since it is declared using Dim.

Public y As Integer

Sub First_Sub()

  x = 10

  y = 20

 Call Third_Sub()

End Sub

Private Sub Second_Sub()

MsgBox "Gone through First, Second and Third Subroutines !"

End Sub
```

Module 2 code

```
Sub Third_Sub()

Debug.Print x

Debug.Print y

Call Second_Sub()

End Sub
```

The two variables x and y that are declared outside a subroutine. This means that their values can carry over into other macros. The variable x has a private scope so only subroutines in the same module can access its value. The variable y has a public scope, meaning that subroutines inside and outside its module can access its value.

The First_Sub() assigns values to x and y and then initiates the Third_Sub().

Third_Sub() can be called even though it is not in the same module because it is a Public Sub.

The Third_Sub() has been designed to display the values of x and y in the immediate window. When you try to print variable x it outputs nothing. This is because x does not exist in Module 2. Therefore, a new variable x is created in Module 2 and since we did not give this new x a value, nothing is printed for the statement Debug.Print x

When we print the value of the variable y, 12 is displayed in the Immediate Window. This is because Module 2 subroutines have access to the public variables declared in Module 1.But the statement "Call Second_Sub()" in the Third_Sub() will result in an error. This is because we are trying to call a private subroutine " Second_Sub" from here. The following changes can be done to avoid this:

1 We could remove the word "Private" from Display_Message

2 We could replace "Private" with "Public" in Second_Sub()

3 We can use the Application level and instead of using Call we could write Application.Run "Second_Sub " (this method serves as an override in case we wanted to keep Second_Sub private for subroutines outside the module.)

Create and Edit Macros

Objective: At the end of this lesson you shall be able t0
- **explain about Macros in VBA.**

Macros offer a powerful and flexible way to extend the features of Excel. They allow the automation of repetitive tasks such as printing, formatting, configuring, or otherwise manipulating data in Excel. In its' simplest form, a macro is a recording of your keystrokes. While macros represent one of the stronger features found in Excel, they are rather easy to create and use. There are six major points that I like to make about macros as follows.

1 **Record, Use Excel, Stop Recording**

To create a macro, simply turn on the macro recorder, use Excel as you normally do, then turn off the recorder. Presto – you have created a macro. While the process is simple from the user's point of view, underneath the covers Excel creates a Visual Basic subroutine using sophisticated Visual Basic programming commands.

2 **Macro Location**

Macros can be stored in either of two locations, as follows:

The workbook you are using, or the Personal Macro Workbook (which by default is hidden from view) If the macro applies to all workbooks, then store it in the Personal Macro Workbook so it will always be available in all of the Excel workbooks; otherwise store it in the current workbook. A macro stored in the current workbook will embedded and included in the workbook, even if you email the workbook to another user.

3 **Assign the Macro to an Icon, Text or a Button**

To make it easy to run your macro, you should assign it to a toolbar icon so it will always be available no matter which workbooks you have open. If the macro applies only to the current workbook, then assign it to Text or a macro Button so it will be quickly available in the current workbook.

4 **Absolute versus Relative Macros**

An "Absolute" macro will always affect the same cells each time whereas a "Relative" macro will affect those cells relative to where the cursor is positioned when invoke the macro. It is crucial that understand the difference.

5 **Editing Macros**

Once created, you can view and/or edit your macro using the View Macros option. This will open the macro subroutine in a Visual basic programming window and provide you with a plethora of VB tools.

6 Advanced Visual Basic Programming

For the truly ambitious CPA, in the Visual Basic Programming window, you have the necessary tools you need to build very sophisticated macros with dialog boxes, drop down menu options, check boxes, radio buttons – the whole works. To see all of this power, turn on the "Developer Tab" in "Excel Options". Presented below are more detailed comments and stepbystep instructions for creating and invoking macros, followed by some example macros.

Page Setup Macro Start recording a new macro called page setup. Select all of the worksheets and then choose Page Setup and customize the header and footers to include page numbers, date and time stamps, file locations, tab names, etc. Assign the macro to an Icon onthe toolbar or Quick Access Bar and insetting headers and footers will be a breeze for the rest of your life.

Print Macros Do you have a template that print frequently from? If so, insert several macro buttons to print each report, a group of reports, and even multiple reports and reporting will be snap in the future.

Delete Data Macro create a macro that visits each cell and erases that data, resetting the worksheet for use in a new set of criteria. Assign the macro to a macro button and will never again have old assumptions mixed in with your newer template

User forms and control in Excel VBA

Objectives: At the end of this lesson you shall be able to
- **define forms and controls in VBA**
- **describe the types of excel forms**
- **describe the properties, methods and events of forms.**

Introduction to Forms and Controls

A form is a document designed with a standard structure and format that makes it easier to enter, organize, and edit information. Forms contain labels, textboxes, drop down boxes and command buttons too.

By using forms and the many controls and objects that you can add to them, you can significantly enhance data entry on your worksheets and improve the way your worksheets are displayed.

Types of Excel forms

There are several types of forms that you can create in Excel: data forms, worksheets that contain Form and ActiveX controls, and VBA UserForms.

Data form

A data form provides a convenient way to enter or display one complete row of information in a range or table without scrolling horizontally. You may find that using a data form can make data entry easier than moving from column to column when you have more columns of data than can be viewed on the screen. Excel can automatically generate a built-in data form for your range or table.

Worksheet with Form and ActiveX controls

A worksheet can be considered to be a form that enables you to enter and view data on the grid.

For added flexibility, you can add controls and other drawing objects to the worksheet, and combine and coordinate them with worksheet cells. For example, you can use a list box control to make it easier for a user to select from a list of items. Or, you can use a spin button control to make it easier for a user to enter a number.

You can display or view controls and objects alongside associated text that is independent of row and column boundaries without changing the layout of a grid or table of data on your worksheet. Many of these controls can also be linked to cells on the worksheet and do not require VBA code to make them work. For example, you might have a check box that you want to move together with its underlying cell when the range is sorted. However, if you have a list box that you want to keep in a specific location at all times, you probably do not want it to move together with its underlying cell.

Creating VBA Forms

A VBA form can be created from the code window. To create a Form in VBA,click on Insert menu in the code window and then click 'UserForm'. A UserForm1 appears in the project window.

When you create or add a form, a module is also automatically created for it. To access the module associated with a form, you can right-click the form and click View Code.Double Clicking on the Form or pressing F7 will also open the Code window. Using Shift F7 will again switch back to the Design Window.

The design time properties of the Form can be set by right clicking on the form and selecting 'Properties'. The same can be achieved by Clicking "F4" or the properties button on the Form.Controls can be placed on the form from the ToolBox as per requirement.

In addition, Controls can be added on the Form programmatically / at run time using the "Add" method. Similarly the controls can be removed from the form at run time / programmatically using the "Remove" method. As an example to add a checkbox control, we can write

Set cb1 = Controls.Add("Forms.CheckBox.1")

Some of the events and methods connected with the form object are:

Events, Activate, Deactivate, Add Control, Remove Control, Click, DblClick, Initialize, KeyPress, Resize, Scroll, Terminate, Zoom etc.

Methods Copy, Paste, Hide, Move, Print Form, Repaint, Scroll, Show etc .

The code needed to perform various operations on Forms is given in Table 1.

A sample Form for data entry of students' details, marks and results is shown in Fig. 1.

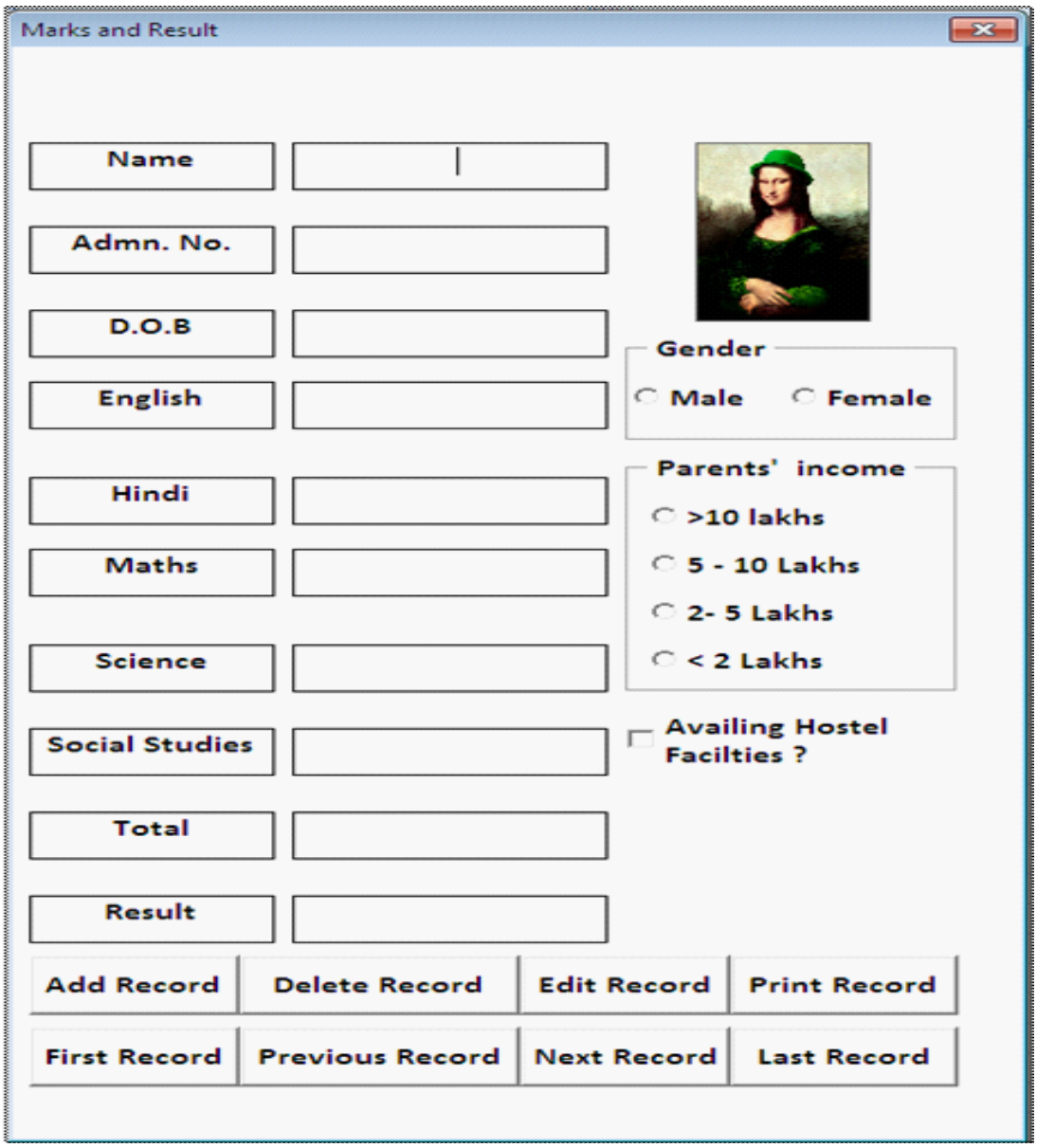
Necessary code can be attached to the Command Buttons and other controls shown. After the user enters the data, the total is calculated and the result is displayed. The records can then be stored appropriately.

Table 1

Userform Application	VBA Code	Action
To Display a UserForm	UserForm1.Show	Displays the UserForm with name UserForm1. This code should be inserted in a Standard VBA Module and not in the Code Module of the UserForm. You can create a button in a worksheet, then right click to assign macro to this button, and select the macro which shows the UserForm.
Load a UserForm into memory but do not display	Load UserForm1	Load statement is useful in case of a complex UserForm that you want to load into memory so that it displays quickly on using the Show method, which otherwise might take a longer time to appear.
Remove a User Form from memory / Close UserForm	Unload UserForm1	Note: The Hide method (UserForm1.Hide) does not unload the UserForm from memory. To unload the UserForm from memory, the Unload method should be used.
	Unload Me	Use the Me keyword in a procedure in the Code Module of the UserForm.
Hide a UserForm	UserForm1.Hide	Using the Hide method will temporarily hide the UserForm, but will not close it and it will remain loaded in memory.
Print a UserForm	UserForm1.PrintForm	The PrintForm method sends the UserForm directly for printing.
Display UserForm as Modeless	UserForm1.Show False	If the UserForm is displayed as Modeless, user can continue working in Excel while the UserForm continues to be shown. Omitting the Boolean argument (False or 0) will display the UserForm as Modal, in which case user cannot simultaneously work in Excel. By default UserForm is displayed as Modal.
Close a UserForm	Unload UserForm1	The Unload method closes the specified UserForm.
	Unload Me	The Unload method closes the UserForm within whose Code Module it resides.
	End	Use the End statement in the "Close" CommandButton to close the form. The "End" statement unloads all forms.
Specify UserForm Caption	UserForm1.Caption = "Bio Data"	Caption is the text which describes and identifies a UserForm and will display in the header of the Userform.
Set UserForm size	UserForm1.Height = 250	Set Height of the UserForm, in points.
	UserForm1.Width = 350	Set Width of the UserForm, in points.
Set UserForm Position:		
Left & Top properties	UserForm1.Left = 30 UserForm1.Top = 50	Distance set is between the form and the Left or Top edge of the window that contains it, in pixels.
Move method	UserForm1.Move 200, 50	Move method includes two arguments which are required - the Left distance and the Top distance, in that order.

Necessary code can be attached to the Command Buttons and other controls shown. After the user enters the data, the total is calculated and the result is displayed. The records can then be stored appropriately.

Fig 1



Methods and Events in VBA

Objectives: At the end of this lesson you shall be able to
- **explain VBA methods and events.**

Methods and Events

Methods

A method is an action you perform with an object. A method can change an object's properties or make the object do something.

For example painting is a Method, building a new room is a method in building a new house.

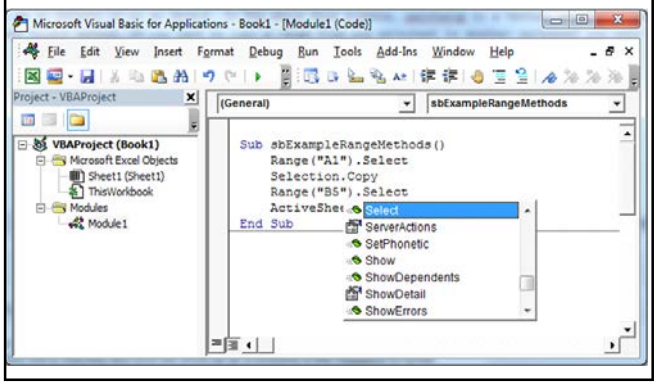
Similarly, if you want to select a range, you need Select method. If you want to copy a range from one worksheet to another worksheet you need Copy method to do it.

The following example Copies the data from Range A1 to B5.

Enter the following code in the Module1 as shown in Fig 1

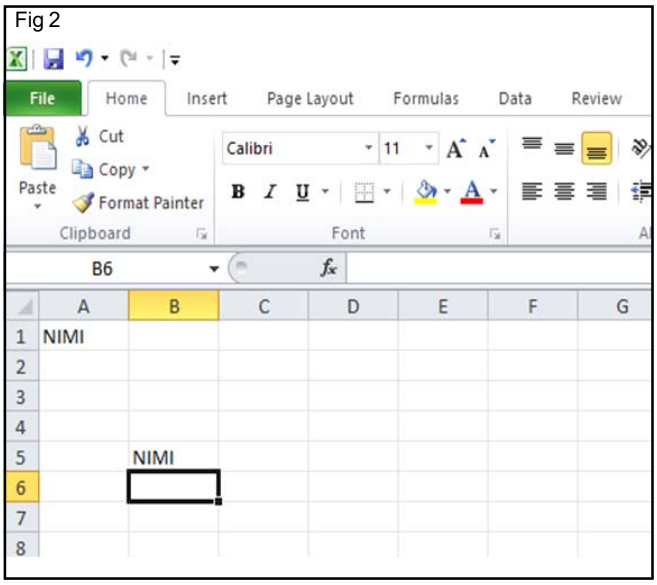
```
Sub sbExampleRangeMethods()
Range("A1").Select
Selection.Copy
Range("B5").Select
ActiveSheet.Paste
End Sub
```

Fig 1



If the above code is executed the content of cell A1 is copied to Cell B5 as shown in the Fig 2.

Fig 2



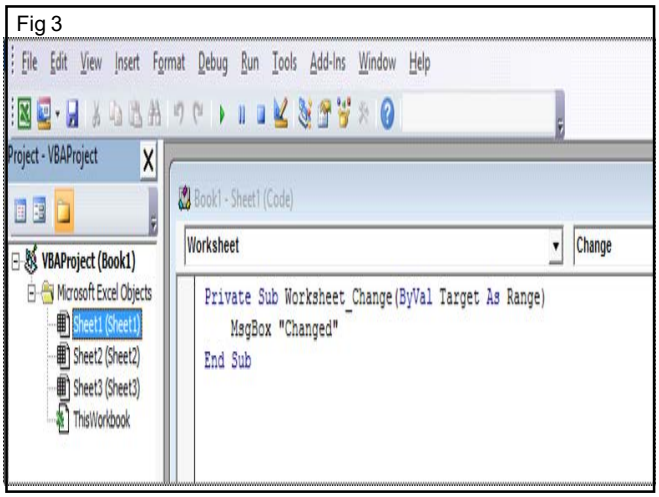
Events

An Event is an action initiated either by user action or by other VBA code. An Event Procedure is a Sub procedure that you write, according to the specification of the event, that is called automatically by Excel when an event occurs. For example, a Worksheet object has an event named Change. If you have properly programmed the event procedure for the Change event, Excel will automatically call that procedure, always named Worksheet_Change and always in the code module of the worksheet, whenever the value of any cell on the worksheet is changed by user input or by other VBA code (but not if the change in value is a result of a formula calculation). You can write code in the Worksheet_Change event procedure to take some action depending on which cell was changed or based upon the newly changed value.

Enter the following code in the Worksheet_Change event as shown in Fig 3.

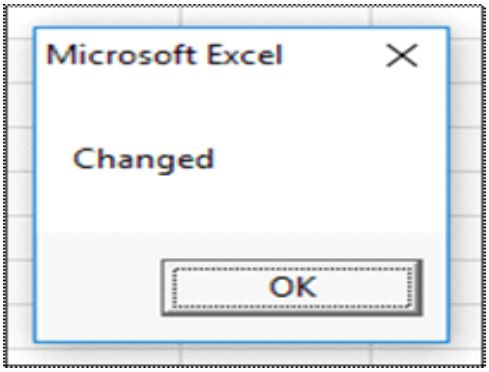
```
Private Sub Worksheet_Change(ByVal Target
As Range)
MsgBox "Changed"
End Sub
```

Fig 3



When we change content of any Cell the following message will be displayed as shown in Fig 4.

Fig 4



Debugging Techniques in VBA

Objectives: At the end of this lesson you shall be able to
- **explain about VBA debugging**
- **explain how to set and clear the Breakpoints**
- **describe use of immediate window**
- **explain about watch window.**

VBA Debugging

In Excel 2010, VBA's debugging environment allows the programmer to momentarily suspend the execution of VBA code so that the following debug tasks can be done:

1 Check the value of a variable in its current state.

2 Enter VBA code in the Immediate window to view the results.

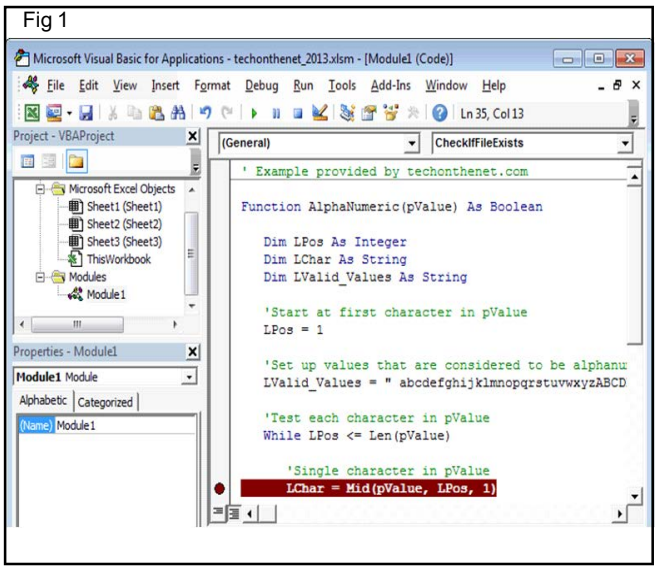
3 Execute each line of code one at a time.

4 Continue execution of the code.

5 Halt execution of the code.

These are just some of the tasks that you might perform in VBA's debugging environment. (Fig 1)

Fig 1



Breakpoint in VBA

In Excel 2010, a breakpoint is a selected line of code that once reached, the program will momentarily become suspended. Once suspended, and to use VBA's debugging environment to view the status of program, step through each successive line of code, continue execution of the code, or halt execution of the code.

And create as many breakpoints in the code as you want. Breakpoints are particularly useful when suspend the program where you suspect a problem/bug exists.

Setting a Breakpoint

First, you need to open the VBA environment. The quickest way to do this is by pressing Alt+F11 while the Excel database file is open.

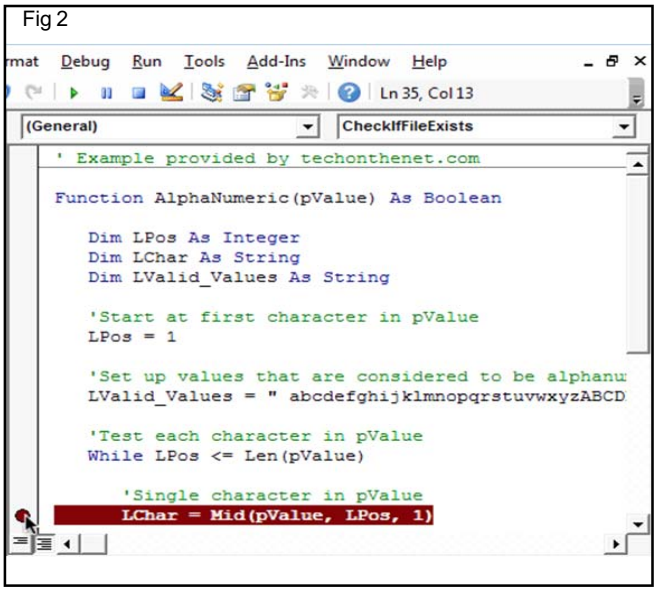
To set a breakpoint, find the line of code where to suspend your program. Left-click in the grey bar to the left of the code. A red dot should appear and the line of code should be highlighted in red.

Clear Breakpoint in VBA

A breakpoint in VBA is indicated by a red dot with a line of code highlighted in red.

To clear a breakpoint in Excel 2010, left-click on the red dot next to the line of code that has the breakpoint. (Fig 2)

Fig 2



In this example, we want to clear the breakpoint at the following line of code:

LChar = Mid(pValue, LPos, 1) (Fig 3)

Now, the breakpoint is cleared and the line of code should look normal again. (Fig 4)

Fig 3

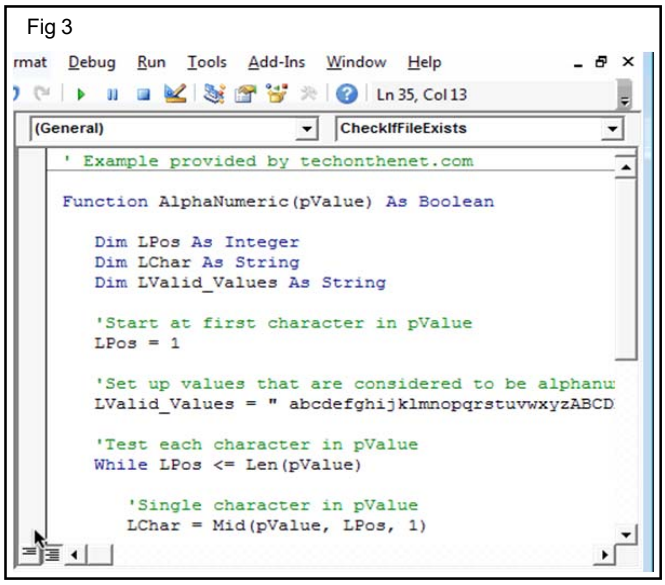
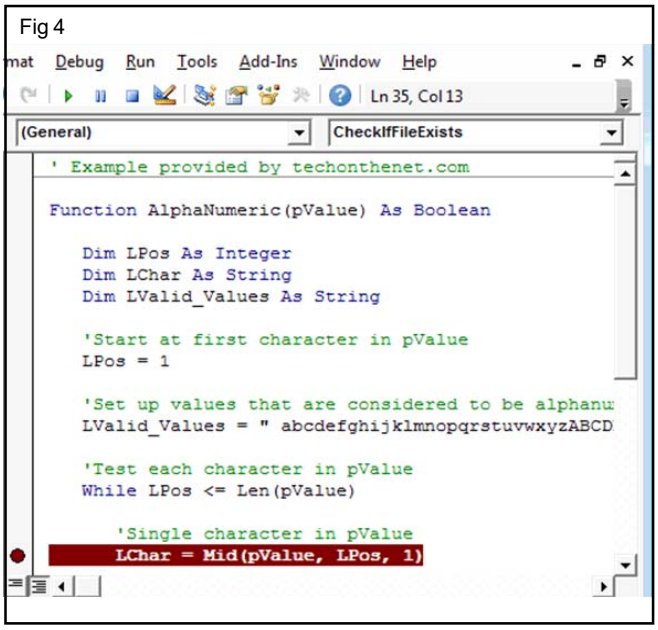


Fig 4



In this example, we've created a breakpoint at the following line of code:

LChar = Mid(pValue, LPos, 1)

Now, the breakpoint is cleared and the line of code should look normal again

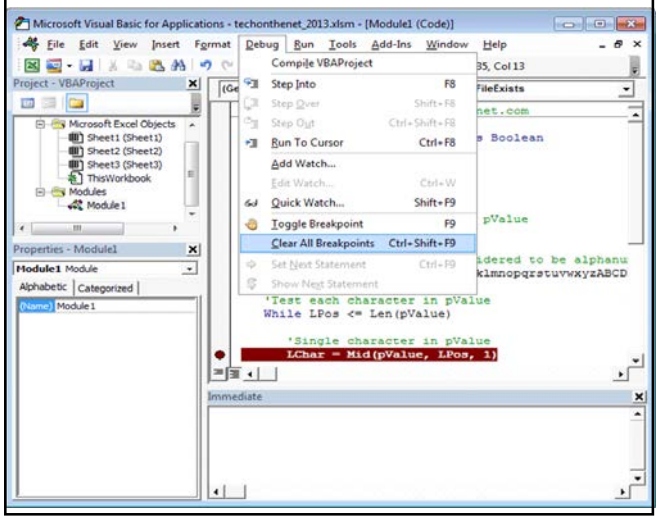
Clearing all Breakpoints

If user use as many breakpoints as you want in Excel 2010, and can save time by clearing all breakpoints in the VBA code at once.

To clear all breakpoints in the program, select "Clear All Breakpoints" under the Debug menu. (Fig 5)

This will remove all breakpoints from the VBA code, so that you don't have to individually remove each breakpoint, one by one.

Fig 5

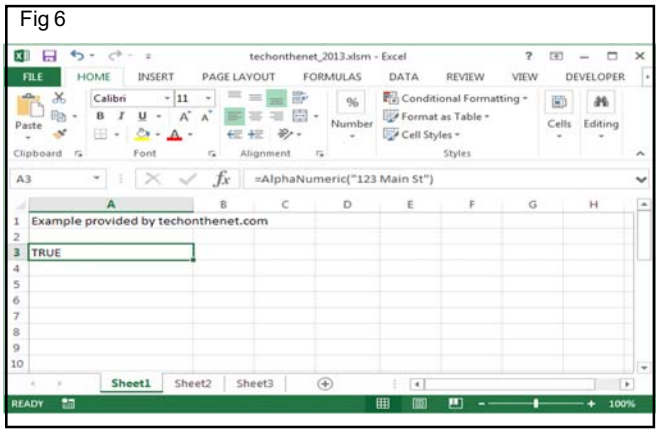


Debug Mode

Now that we know how to set and clear breakpoints in Excel 2010, let's take a closer look at the debug mode in VBA.

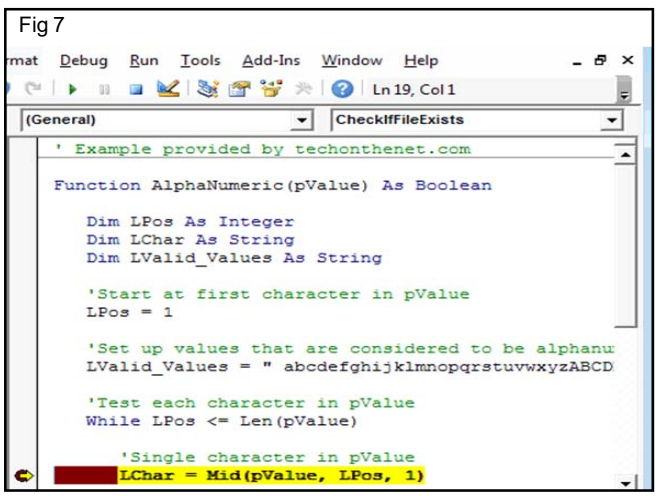
In our example, we've set our breakpoint and entered our AlphaNumeric function as a formula in a cell. This will cause the VBA code to execute. (Fig 6)

Fig 6



When the breakpoint is reached, Excel will display the Microsoft Visual Basic window and highlight the line (in yellow) where the code has been suspended. (Fig 7)

Fig 7



Now we are in debug mode in our Excel spreadsheet. Now we can do any of the following:

1 Check the value of a variable in its current state.

2 Enter VBA code in the Immediate window to view the results.

3 Execute each line of code one at a time.

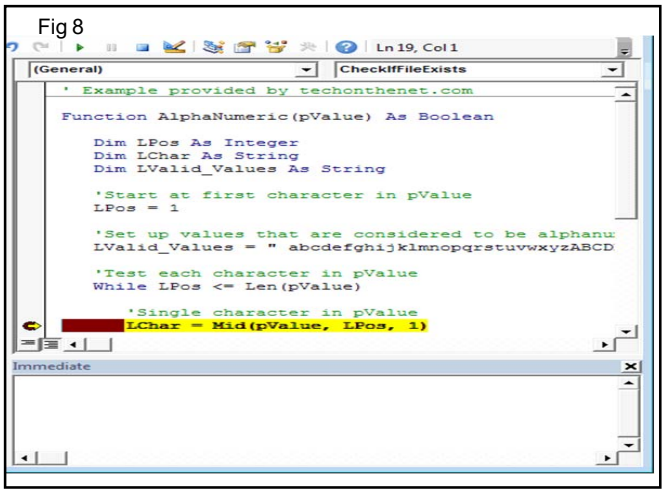
4 Continue execution of the code.

5 Halt execution of the code.

Using the Immediate Window

In Excel 2010, the Immediate window can be used to debug your program by allowing you to enter and run VBA code in the context of the suspended program. (Fig 8)

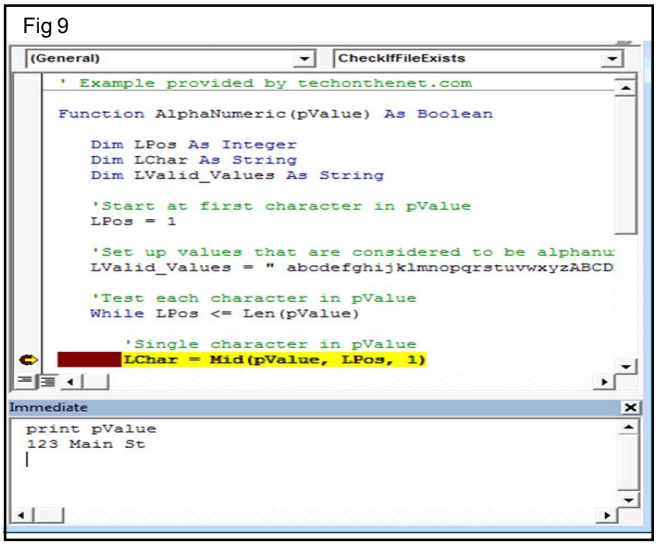
Fig 8



We've found the Immediate window to be the most help when we need to find out the value of a variable, expression, or object at a certain point in the program. This can be done using the print command.

For example, if you wanted to check the current value of the variable called pValue, you could use the print command as follows: (Fig 9)

Fig 9



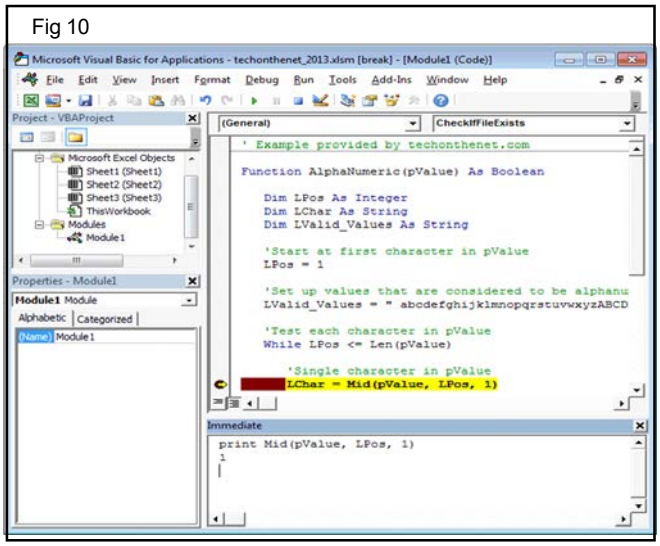
In this example, we typed **print pValue** in the Immediate window and pressed ENTER.

Print pValue

The Immediate window displayed the result in the next line. In this case, the print pValue command returned **123 Main St**.

You can also type more complicated expressions in the Immediate window. (Remember to press ENTER.) For example: (Fig 10)

Fig 10



In this example, we typed **print Mid(pValue, LPos, 1)** in the Immediate window and pressed ENTER.

print Mid(pValue, LPos, 1)

The Immediate window displayed the result of **1** in the next line.

The Immediate window can be used to run other kinds of VBA code, but bear in mind that the Immediate window can only be used when debugging so any code that you run is for debugging purposes only. The code entered in the Immediate window does not get saved and added to the existing VBA code

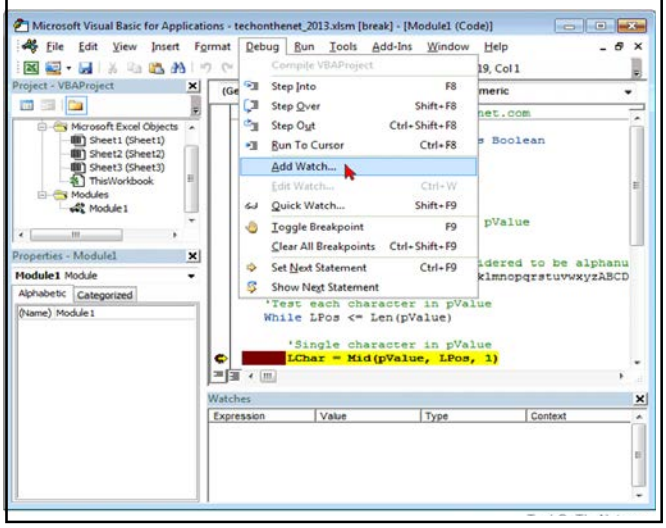
Adding a Watch Expression

The Watch Window displays the value of a watched expression in its current state. This can be extremely useful when debugging VBA code. Let's explore how to add an expression to the Watch Window.

To add a Watch expression, select **Add Watch** under the **Debug** menu. (Fig 11)

When the *Add Watch* window appears, enter the expression to watch and click the OK button when you are done. (Fig 12)

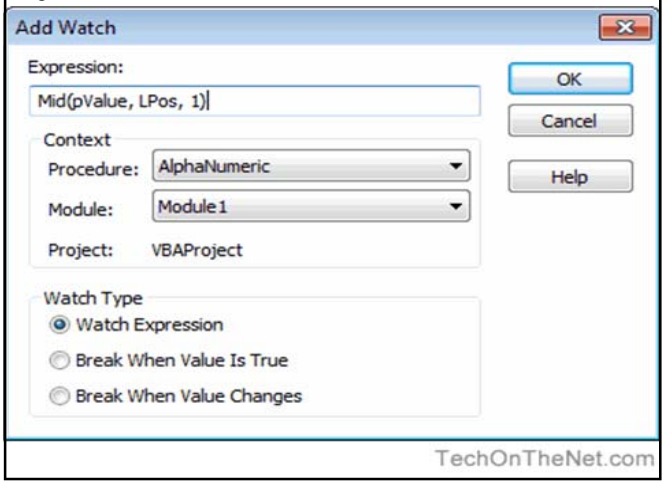
Fig 11



In this example, we've entered the following watch expression in the **Expression** field:

Mid(pValue, LPos, 1)

Fig 12



Next, we've selected AlphaNumeric as the **Procedure** and Module1 as the **Module** when setting up the **Context** for the watched expression.

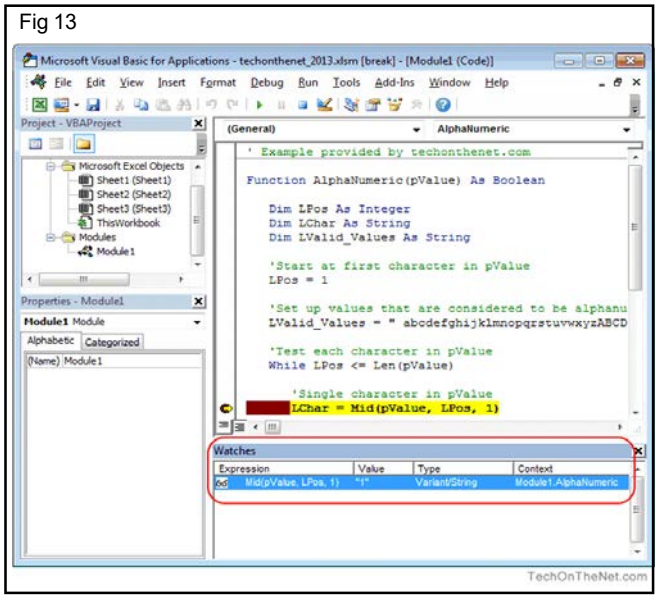
Finally, we've selected *Watch Expression* as the **Watch Type** but there are 3 options to choose from:

Watch Type	Description
Watch Expression	To display the value of the watched expression in its current state
Break When Value Is True	To stop the execution of the code when the value of the watched expression is True
Break When Value Changes	To stop the execution of the code when the value of the watched expression changes

When return to the VBA window, the Watch Window will automatically appear if it was previously hidden. Within the Watch Window, all of the watched expressions should be listed including the one that we just added. (Fig 13)

As you can see, the expression Mid(pValue, LPos, 1) now appears in the Watch Window with a value of "1". Adding a watch is a great way to keep track of variables or expressions of interest when debugging the VBA code.

Fig 13



Object Oriented Programming concepts, Concepts of classes, Objects, properties and Methods

Objectives: At the end of this lesson you shall be able to
- **explain Class and objects and its features**
- **explain VBA Class modules Versus VBA normal modules**
- **list out parts of a class module and its properties**
- **explain class module events.**

Introduction

VBA Class Modules allow the user to create their own objects. In languages such as C# and Java, **classes** are used to create objects. **Class Modules** are the VBA equivalent of these classes. The major difference is that VBA Class Modules have a very limited type of Inheritance* compared to classes in the other languages. In VBA, Inheritance works in a similar way to Interfaces in C#\Java.

In VBA we have built-in objects such as the Collection, Workbook, Worksheet and so on. The purpose of VBA Class Modules is to allow us to custom build our own objects.

Let's start this post by looking at why we use objects in the first place.

Inheritance is using an existing class to build a new class.
Interfaces are a form of Inheritance that forces a class to implement specifics procedures or properties.

Objects

Using objects allows us to build our applications like we are using building blocks.

The idea is that the code of each object is self-contained. It is completely independent of any other code in our application.

Advantages of Using Objects

Treating parts of our code as blocks provide us with a lot of advantages

1 It allows us to build an application one block at a time.

2 It is much easier to test individual parts of an application.

3 Updating code won't cause problems in other parts of the application.

4 It is easy to add objects between applications.

Disadvantages of Using Objects

With most things in life there are pros and cons. Using VBA class modules is no different. The following are the disadvantages of using class module to create objects

1 It takes more time *initially* to build applications*.

2 It is not always easy to clearly define what an object is.

3 People new to classes and objects can find them difficult to understand at first.

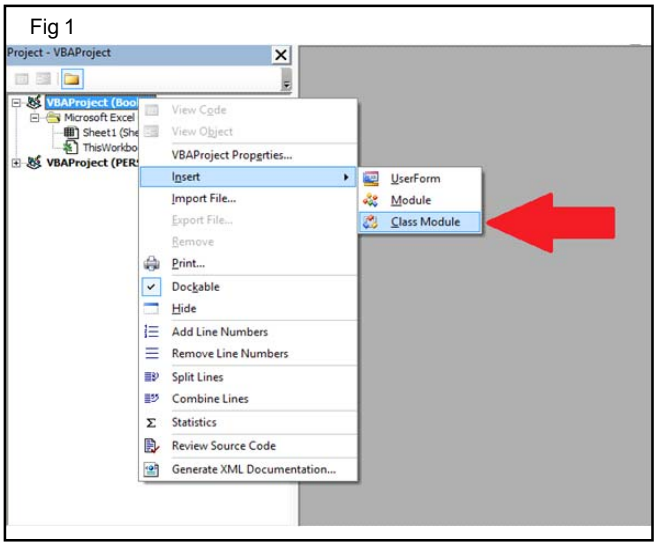
If create an application using objects it will take longer to create it initially have to spend more time planning and designing it. However, in the long run it will save a huge amount of time. The code will be easier to manage, update and reuse.

Creating a Simple Class Module

Let's look at a very simple example of creating a class module and using it in our code.

To create a class module we right-click in the Project window and then select **Insert** and **Class Module.** (Fig 1)

Fig 1



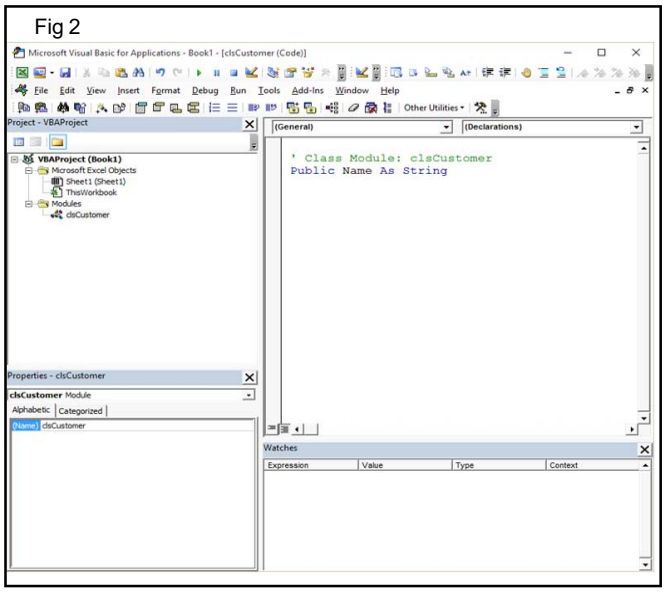
Adding a Class Module

Our new class is called **Class1**. We can change the name in the Properties window.

Let's change the name of the class module to **clsCustomer**. Then we will add a variable to the class module like this

Public Name **AsString (Fig 2)**

Fig 2



We can use now use this class module in any module(standard or class) in our workbook. For example

' Create the object from the class module

Dim oCustomer **AsNew** clsCustomer

' Set the customer name

oCustomer.Name = "John"

' Print the name to the Immediate Window(Ctrl + G)

Debug.Print oCustomer.Name

Class Module versus Objects

People who are new to using classes and VBA class modules, often get confused between what is a class and what is an object.

Let's look at a real-world example. Think of a mass-produced item like a coffee mug. A design of the mug is created first. Then, thousands of coffee mugs are created from this design.

This is similar to how class modules and objects work.

The **class module** can be thought of as the design.

The **object** can be thought of as the item that is created from the design.

The **New** keyword in VBA is what we use to create an object from a class module. For example

' Creating objects using new

Dim oItem **AsNew** Class1

Dim oCustomer1 **AsNew**clsCustomer

Dim coll **AsNew** Collection

Note: We don't use New with items such as Workbooks and Worksheets. See When New is not required for more information.

VBA Class Modules Versus VBA Normal Modules

Writing code in a class module is almost the same as writing code in a normal module. We can use the same code we use in normal modules. It's how this code is used which is very different.

Let's look at the two main differences between the class and normal module. These often cause confusion among new users.

Difference 1 – How the modules are used

If want to use a sub/function etc. from a class module must create the object first.

For example, imagine we have two identical **PrintCustomer** subs. One is in a class module and one is in a normal module…

' CLASS MODULE CODE - clsCustomer

Public Sub PrintCustomer()

Debug.Print "Sample Output"

End Sub

' NORMAL MODULE CODE

Public Sub PrintCustomer()

Debug.Print "Sample Output"

End Sub

You will note the code for both is exactly the same.

To use the **PrintCustomer** sub from the class module, you must first create an object of that type

' Other Module

Sub UseCustomer()

Dim oCust **AsNew** clsCustomer

oCust.PrintCustomer

EndSub

To use **Print Customer** from the normal module you can call it directly

‘ Other Module

Sub Use Customer()

Print Customer

End Sub

Difference 2 – Number of copies

Whencreate a variable in a normal module there is only one copy of it. For a class module, there is one copy of the variable for each object you create.

For example, imagine we create a variable **Student Name** in both a class and normal module..

‘ NORMAL MODULE

Public StudentName **As String**

‘ CLASS MODULE

Public Studen tName **As String**

For the normal module variable there will only be one copy of this variable in our application.

StudentName = “Ram”

For the class module a new copy of the variable **Student Name** is created each time a new object is created.

Dim student1 **As New** clsStudent

Dim student 2 **As New** clsStudent

student1.Student Name = “Bill”

student2.Student Name = “Ted”

When fully understand VBA class modules, these differences will seem obvious.

The Parts of a Class Module

There are four different items in a class module. These are

1 **Methods** – functions/subs.

2 **Member variables** – variables.

3 **Properties**– types of functions/subs that behave like variables.

4 **Events** – subs that are triggered by an event.

And can see they are all either functions, subs or variables.

Let’s have a quick look at some examples before we deal with them in turn

‘ CLASS MODULE CODE

‘ Member variable

Private dBalance **As Double**

‘ Properties

Property Get Balance () **AsDouble**

Balance = dBalance

EndProperty

Property Let Balance(dValue**As** Double)

dBalance = dValue

End Property

‘ Event - triggered when class created

Private Sub Class_Initialize()

dBalance = 100

EndSub

‘ Methods

Public Sub Withdraw (dAmount**As** Double)

dBalance = dBalance - dAmount

End Sub

Public Sub Deposit (dAmount**As** Double)

dBalance = dBalance + dAmount

EndSub

Now that we have seen examples, let’s look at each of these in turn.

Class Module Methods

Methods refer to the procedures of the class. In VBA procedures are subs and functions. Like member variables they can be Public or Private.

Let’s look at an example

‘ CLASS MODULE CODE

‘ Class name: clsSimple

‘ Public procedures can be called from outside the object

Public Sub PrintText (sText**As** String)

Debug.PrintsText

EndSub

Public Function Calculate (dAmount**As** Double) **As Double**

Calculate = dAmount - GetDeduction

End Function

‘ private procedures can only be called from within the Class Module

Private Function GetDeduction () **As Double**

GetDeduction = 2.78

EndFunction

We can use the **clsSimple** class module like this

Sub Class Members ()

Dim oSimple **As New** clsSimple

oSimple.PrintText “Hello”

Dim dTotal **As Double**

dTotal = oSimple.Calculate(22.44)

Debug.Print dTotal

EndSub

Class Module Member Variables

The member variable is very similar to the normal variable we use in VBA. The difference is we use **Public** or **Private** instead of **Dim**.

‘ CLASS MODULE CODE

Private Balance **AsDouble**

Public AccountID **As String**

Note: Dim and Private do exactly the same thing but the convention is to use Dim in sub/ functions and to use Private outside sub/ functions.

The **Public** keyword means the variable can be accessed from outside the class module. For example

Dim oAccount **AsNew** clsAccount

‘ Valid - AccountID is public

oAccount.AccountID = “499789”

‘ Error - Balance is private

oAccount.Balance = 678.90

In the above example we cannot access **Balance** because it is declared as **Private**. We can only use a **Private** variable within the class module. We can use in a function/ sub in the class module e.g.

‘ CLASS MODULE CODE

Private Balance **As Double**

Public Sub SetBalance()

Balance = 100

Debug.Print Balance

End Sub

It is considered poor practice to have public member variables. This is because the code allowing outside the object to interfere with how the class works. The purpose of the using classes is so that hide what is happening from the caller.

To avoid the user directly talking to the member variables we use Properties.

Class Module Properties

1 **Get** – returns an object or value from the class

2 **Let** – sets a value in the class

3 Set – sets an object in the class

Format of VBA Property

The normal format for the properties are as follows:

Public Property Get () **AsType**

End Property

Public Property Let (varname**AsType**)

End Property

Public PropertySet (varname**AsType**)

EndProperty

We have seen already that the Property is simply a type of sub. The purpose of the Property is to allow the caller to get and set values.

Use of Properties

Imagine we have a class that maintains a list of Countries. We could store the list as an array

‘ Use array to store countries

Public arrCountries **As Variant**

```
‘ Set size of array when class is initialized

Private Sub Class_Initialize()

ReDim arrCountries (1 To 1000)

End Sub
```

When the user wants to get the number of countries in the list they could do this

```
‘ NORMAL MODULE CODE

Dim oCountry As New clsCountry

‘ Get the number of items

NumCountries = UBound(oCountry.arrCountries) + 1
```

There are two major problems with the above code

1. To get the number of countries you need to know how the list is stored e.g. Array.
2. If we change the Array to a Collection, we need to change all code that reference the array directly.

To solve these problems we can create a function to return the number of countries

```
‘ CLASS MODULE CODE - clsCountryList

‘ Array

Private arrCountries () As String

Public Function Count () AsLong

   Count = UBound(arrCountries) + 1

End Function
```

We then use it like this

```
‘ MODULE CODE

Dim oCountries As New clsCountries

Debug.Print “Number of countries is “ &oCountries.Count
```

This code solves the two problems we listed above. We can change our Array to a Collection and the caller code will still work e.g.

```
‘ CLASS MODULE CODE

‘ Collection

Private collCountries() As Collection

Public FunctionCount() AsLong

   Count = collCountries.Count

End Function
```

The caller is oblivious to how the countries are stored. All the caller needs to know is that the **Count** function will return the number of countries.

As we have just seen, a sub or function provides a solution to the above problems. However, using a **Property** can provide a more elegant solution.

Using a Property instead of a Function/Sub

Instead of the creating a **Count** Function we can create a **Count** Property. As you can see below they are very similar

```
‘ Replace this

Public Function Count() As Long

   Count = UBound(arrCountries) + 1

End Function

‘ With this

Property Get Count () As Long

   Count = UBound(arrCountries) + 1

End Function
```

In this scenario, there is not a lot of difference between using the Property and using a function. However, there are differences. We normally create a **Get** and **Let** property like this

```
‘ CLASS MODULE CODE - clsAccount

Privated TotalCost As Double

Property Get TotalCost () As Long

Total Cost= dTotalCost

End Property

Property Let Total Cost (dValue As Long)

dTotal Cost = dValue

End Property
```

Using **Let** allows us to treat the property like a variable. So we can do this

```
oAccount.Total Cost = 6
```

The second difference is that using **Let** and **Get** allows us to use the same name when referencing the Get or Let property. So we can use the property like a variable. This is the purpose of using Properties over a sub and function.

oAccount.TotalCost = 6

dValue = oAccount.TotalCost

If we used a function and a sub then we cannot get the behaviour of a variable. Instead we have to call two different procedures e.g.

oAccount.SetTotalCost 6

dValue = oAccount.GetTotalCost

You can also see that when we used Let we can assigned the value like a variable. When we use **Set Total Cost**, we had to pass it as a parameter.

The Property in a Nutshell

1 The Property hides the details of the implementation from the caller.

2 The Property allows us to provide the same behaviour as a variable.

Types of VBA Property

There are three types of Properties. We have seen Get and Let already. The one we haven't looked at is **Set**.

Set is similar to **Let** but it is used for an object(see Assigning VBA Objects for more detail about this).

Originally in Visual Basic, the **Let** keyword was used to assign a variable. In fact, we can still use it if we like.

‘ These line are equivalent

Let a = 7

a = 7

So we use **Let** to assign a value to a variable and we use **Set** to assign an object to an object variable

‘ Using Let

Dim a **As Long**

Let a = 7

‘ Using Set

Dim coll1 **As** Collection, coll2 **As** Collection

Set coll1 = **New** Collection

Set coll2 = coll1

- **Let** is used to assign a value to a basic variable type.
- **Set** is used to assign an object to an object variable.

In the following example, we use **Get** and **Let** properties for a string variable

‘ CLASS MODULE CODE

‘ SET/LET PROPERTIES for a variable

Private m_sName **As String**

‘ Get/Let Properties

Property Get Name() **As String**

Name = m_sName

End Property

Property Let Name (sName**As** String)

m_sName = sName

End Property

We can then use the **Name** properties like this

Sub Test Let Set()

Dim sName **As String**

Dim coll **As New** Collection

Dim oCurrency **As New** clsCurrency

‘ Let Property

oCurrency.Name = “USD”

‘ Get Property

sName = oCurrency.Name

End Sub

In the next example, we use **Get** and **Set** properties for an object variable

‘ CLASS MODULE CODE

Private m_collPrices **As** Collection

‘ Get/Set Properties

Property Get Prices() **As** Collection

Set Prices = m_collPrices

End Property

Property Set Prices (collPrices**As** Collection)

Set m_collPrices = collPrices

End Property

We can then use the properties like this

Sub Test Let Set ()

Dim coll1 **As New** Collection

Dim oCurrency **As New** cls Currency

‘ Set Property

Set oCurrency.Prices = coll1

‘ Get Property

Dim coll2 **As** Collection

Set Coll2 = oCurrency.Prices

EndSub

We use the Get property to return the values for both items. Notice that even though we use the **Get** Property to return the Collection, we still need to use the **Set** keyword to assign it.

Class Module Events

A class module has two events

1 **Initialize** – occurs when a new object of the class is created.
2 **Terminate** – occurrs when the class object is deleted.

In Object Oriented languages like C++, these events are referred to as the **Constructor** and the **Destructor**. In most languages, you can pass parameters to a constructor but in VBA you cannot. We can use a **Class Factory** to get around this issue as we will see below.

Initialize

Let’s create a very simple class module called clsSimple with **Initialize** and **Terminate** events

‘ CLASS MODULE CODE

Private Sub Class_Initialize()

Msg Box “Class is being initialized”

End Sub

Private Sub Class_Terminate()

Msg Box “Class is being terminated”

End Sub

Public Sub Print Hello ()

Debug.Print “Hello”

End Sub

In the following example, we use **Dim** and **New** to create the object.

In this case, **oSimple** is not created until we reference it for the first time e.g.

Sub Class Event sInit2 ()

Dim oSimple **As New** clsSimple

‘ Initialize occurs here

oSimple.PrintHello

EndSub

When we use **Set** and **New** together the behaviour is different. In this case the object is created when **Set** is used e.g.

Sub Class Events Init()

Dim oSimple **As** clsSimple

‘ Initialize occurs here

Set oSimple = **New** clsSimple

oSimple.PrintHello

End Sub

Note: For more information about the different between using New with Dim and using New with Set see Subtle Differences of Dim Versus Set

As said earlier, you cannot pass a parameter to **Initialize**. If you need to do this you need a function to create the object first

‘ CLASS MODULE - clsSimple

Public Sub Init (Price **As** Double)

EndSub

‘ NORMAL MODULE

PublicSubTest()

‘ Use CreateSimpleObject function

Dim oSimple **As** clsSimple

Set oSimple = CreateSimpleObject(199.99)

End Sub

Public Function CreateSimpleObject(Price **As** Double) **As** clsSimple

Dim oSimple **As New** clsSimple

oSimple.Init Price

Set CreateSimpleObject = oSimple

End Function

We will expand on this CreateSimpleObject in Example 2 to create a **Class Factory**.

Terminate

The Terminate event occurs when the class is deleted. This happens when we set it to **Nothing**

Sub Class EventsTerm ()

Dim oSimple **As** clsSimple

Set oSimple = **New**clsSimple

‘ Terminate occurs here

Set oSimple = **Nothing**

End Sub

If we don’t set the object to **Nothing** then VBA will automatically delete it when it goes out of scope.

What this means is that if we create an object in a procedure, when that procedure ends VBA will delete any objects that were created.

Sub Class EventsTerm2()

Dim oSimple **As New** clsSimple

‘ Initialize occurs here

oSimple.PrintHello

‘ oSimple is deleted when we exit this Sub calling Terminate

EndSub

Introduction to Tally, Features and Advantages

Objectives: At the end of this lesson you shall be able to
- **explain about the tally software & history of Tally**
- **features of the tally software**
- **advantages of the tally software.**

Introduction to Tally : Tally is an complete accounting software. It is a versatile and massive software package, being used by various types of business Organisations.

History of Tally

Tally is a complete business solution for any kind of Business Enterprise. It is a full fledged accounting software.

The Initial Release of Tally was Tally 4.5 version. This is DOS (MS-DOS) based software released in the beginning of 1986's. It had Basic Financial Accounting / Book Keeping Tools. Personal computers had gaining popularity in India those days.

Peutronics (The Company that develops Tally) used this opportunity and put their Tally Version 4.5 on the market.

Auditors and Accountants who used to maintain large volumes of hard-bound notebooks were amazed at the ability of Tally to calculate Balance sheets and Profit Loss accounts within seconds. All you need to do is just create Ledgers and enter vouchers. Tally will do the rest. It will create all the statements, Trial Balance and Balance Sheet For you.

The subsequent Tally releases are Tally 5.4, Tally 6.3, Tally 7.2, Tally 8.1 and Tally 9.0, Tally ERP (Enterprise Resources Planning). These release Include support for Inventory used to stock maintenance of the company, Payroll which used to employee salary calculation and wages payments and Multi Lingual support in Many Indian languages Hindi, Tamil, Telugu, Kannada, Malayalam, Gujarati, Marathi and more.

Versions of Tally:

Tally 4.0 & Tally 4.5: This version MS-DOS support financial accounting system. It takes care of accounting activities only such as Ledgers Classification Vouchers Entry. It provides simple financial reports and bill wise analysis of debtors and creditors in the business.

Tally 5.0: This version is an upgraded version to tally 4.5 and it works in windows operating system Inventory modules is introduced in this version, which involves detailed inventory, structure invoicing and integrating accounting and Inventory records.

Tally 5.4: This version is an improved module over the version 5.0 where it is capable of converting earlier data formats in to the current data format. This is possible though Import of Data Facility.

Tally 6.3: Tally 6.3 is extended enterprise systems whereby it interacts with other system through ODBC (Open Data Base Connectivity) you and e-mail upload your financial records form tally.

Tally 7.2: This version is an integrated enterprise system provides different kind of taxes like VAT, TDS & TCS and Service Tax modules is introduced in this version.

Tally 8.1: Tally 8.1 is multi language support software. It supports 10 Languages includes is introduced in this version.

Tally 9.0: This version is an improved model over the version 8.1. it supports 13 Languages (Includes Foreign Languages). Payroll, POS (Point of Sales) modules is introduced in this version.

Tally.ERP9: This is the latest version which provides different features like remote access,much powerful data security, tally.net and many more.

Tally ERP9 is considered as the latest version.

Features of Tally

1 Accounting Features

i Handles different types of vouchers
- Payments Receipt
- Journals
- Debit Notes
- Credit Notes
- Sales Notes
- Purchase Notes
- Receipt Notes
- Delivery Notes etc.

ii Handles Primary Books of Accounts
- Cash Book
- Bank Book
- Ledger

- Purchase registers
- Sales Registers etc.,

iii Used to prepare Statement of Accounts

- Trial Balance
- Profit and loss Accounts
- Trade Accounts
- Balance Sheet
- Funds Flow
- Cash Flow

2 Financial Management Features

- We can get closing stock value as entered in stock ledgers by non-integrating Accounts and Inventory.
- Daily Balances and Transactions value can be got.
- Funds flow and cash flow statements to track movement of cash and funds in the company.
- Tally computes interests as per book date.
- Tally provides Budgeting option.
- Ratio Analysis provides important performance ratios that give the pulse of the corporate health.

3 Inventory Management Features

- Flexible invoicing and billing terms.
- Flexible units of measure.
- Stock Transfer-Tally provides stock journal.
- Stock query provides all relevant information for any stock item in a single screen.
- Multiple stock valuation methods like FIFO, LIFO and average methods are enhanced in Tally.

4 Security Features

- The system administrator can define multiple levels of security. Hence can credit, authorize-users, assign passwords and assign specific task rights.
- Tally offers data encryption (Tally vault) and follows Data Encryption Standard (DES) Encryption methods.
- Tally provides options for data backup in floppy/hard disk and restoration of backup data.
- Tally locker: It is a small portable hardware device of a thumb size with storage capacity of 16MB. We can store data and work directly at tally locker. When the task is over, we can simply keep it in our pocket. It is also used for backup.

5 Technological features

- Tally allows importing data from other software as well as exporting data from tally.
- ODBC connectivity is available in Tally. We can connect applications like MS Word, MS Excel, Oracle and can use data from tally directly.
- While working with tally, we can e-mail, browse a website. We can send a report on document directly from tally.
- We can upload reports on the website directly from tally.
- Protocol support for HTTP, HTTPS, FTP, SMTP, ODBC and RAW sockets with data interchange formats like XML, HTML, related formats.

Advantages of the Tally

1 Simple and Rapid Installation

- Tally.ERP9's installation is a wizard driven, simple and speedy process involving minimal user-intervention. The software occupies tiny space and can be installed on any drive. Tally.ERP9 supports installation on multiple systems connected to a network with different operating systems (Windows98, NT, 2000, XP and Windows7)

2 Auto Backup and Restore

- Tally.ERP9 provides automatic backup facility to secure your company from any kind of data loss / corruption and helps in smooth functioning of your business. Tally.ERP9 safeguards your data from any loss due to power failure or improper shutdown of the system.

3 Tally Audit

- Tally.ERP9 audit feature allows you to verify, validate and accept accounting information based on the masters, users and transactions (vouchers).

4 Split Company Data

- Tally.ERP9 allows splitting of company data into multiple companies for the required financial period. Once the data is split, the closing balances of the previous period are automatically carried forward as the opening balance for the subsequent period.

5 Import and Export of Data

- Tally.ERP9 allows you to flexibly export and import data in various formats such as MS EXCEL, JPEG, PDF, XML, HTML or ASCII format.

6 Graphical Analysis of Data

- Tally.ERP9 allows easy analysis of results / reports with graphical representation of values.

7 Duties and Taxes

- Tally.ERP9 allows Statutory Reporting for VAT (Value Added Tax), CST (Central Sales Tax), Service Tax, TCS (Tax Collected all Source), TDS (Tax Deducted at Source), FBT (Fringe Benefit Tax), GST (Goods and Service Tax).

8 E-Mail Facility

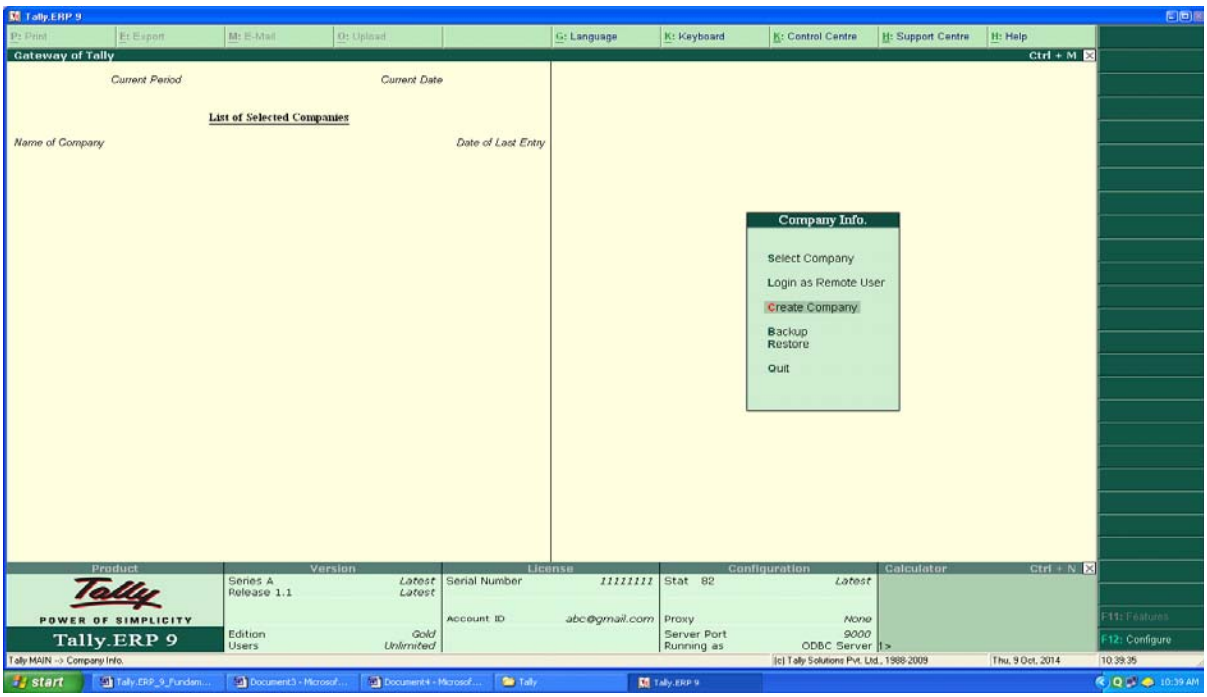
- Tally.ERP9 supports mailing of required information to intended recipients and also mass mailing facility for certain reports like Payslip etc.

Implementing Accounts in Tally - Basics of Accounting, Golden Rules of Accounting, Voucher Entry, Ledger Posting, Final Accounts Preparation

Objectives: At the end of this lesson you shall be able to
- **explain the tally software screen**
- **understand accounting, its types, terms used**
- **understand golden rules of Accounting**
- **state about the journals, ledgers with entry posting methods**
- **prepare cash book, bank book etc.**
- **know shortcut keys.**

Tally Software : In Gateway of Tally the following option is display. (Fig 1)

Fig 1



Screen of the Tally

Tally Screen: Tally screen can be divided into following four broad parts.

1 Product info

2 Button bar

3 Calculator

4 Work area

Product info: Product info bar in Tally.ERP9 consisting of the information about the product.

1 Developer company

2 Software version and Release

3 Software Serial Number

4 Single or Multiple User

5 System Day and Date

6 System Time

Button Bar: Buttons appearing in the button bar (at right of the screen) provide quick access to different options. Buttons on the button bar is context sensitive, different buttons appear at different screens.

Calculator: By default the work area becomes active and calculator remains inactive. Press Ctrl + N that would activate the calculator when calculator is active, we can enter value and operators.

The calculator follows BODMAS rule that indicates the execution sequence: Bracket, Power, Division, Multiplication, Addition and Subtraction.

Work area

The work at Gateway is broadly separated into two sections.

1 The right hand side contains the popup menu, where we would select instructions to Tally.

2 The left hand side displays list of selected companies.

On left part of the screen

Current Period

Financial period with which we are working is displayed for reference. To change the Financial period click "F2:period" button in the button bar or press <Alt> + <F2>.

Current Date

It is not the calendar date but the date we worked last during the current period. Vouchers will have the same date as of current date. To change current date click "F2:Date" button in the button bar or press F2.

List of selected companies

Name of all selected companies with last date voucher entry is displayed here. If we select more than one company, the active company is shown at the top of list in bold face and others appear next in normal fonts.

On right part of the screen

Gateway of Tally menu

Primary choices of menu is displayed. The menu operations are dependent on selection/activation of operations. In some cases, we may find some operations grayed indication that the option is not active.

For example, upon selecting a company maintaining Accounts only, Inventory Info and stock summary options would be grayed.

Selections of menu

We can select a menu in either of the following mode:

1 Press the highlighted character. For example, to select create company, press 'C' the highlighted character.

2 Move highlight bar on the option with Up/Dn arrow key and press<Enter>.

3 Using mouse double click the option. (We have to click twice very fast).

Company Info Menu

On loading Tally, by default, Company Info menu appear to select a company we wish to work with. If we are installing Tally first time. select company option would remain inactive and by default cursor would appear on Create Company option.

Buttons at gateway

Help (Hotkey: Alt+H): This button launches "Tally Reference Manual. This is a compiled HTML Help file. Normally on clicking this button, the relevant content in respect of the screen would be displayed. If no context sensitive Help exists, contents would be displayed. We can select the topic we wish to view.

Web browser (Hot key: Alt+W)

This button launches the default installed web browser. For example, internet explorer is the default web browser, on clicking the button IE will be loaded. The browser will appear within the work area of the tally screen. All other areas of Tally screen still remains in the screen.

F1- Select cmp (Hoy key: F1)

This button will display the list of companies. Move the highlight bar and press <Enter> key to select a particular company. Or simply press F1 key on the keyboard.

Introduction to Accounting

It is the language of business through which normally a business house communicates with the outside world.

Accounting may be defined as "the art of recording, classifying summarising and analysing transactions in the books of accounts".

Accounting and accountancy are often synonymous terms.

Accounting can be classified into two parts.

1 Financial accounting

2 Management Accounting

Financial Accounting

The accounting for revenues, expenses assets and liabilities that is commonly carried on in the general offices of a business is termed as Financial Accounting.

Its aim is to ascertain the profit or loss and to state the financial position of the business as at a particular date.

Scope of Financial Accounting

1 Recording information in account books like journals, cashbooks etc

2 Classifying the accounts using ledger.

3 Summarising the information as statements like (1) Trial balance (2) Income statement and (3) Balance sheet

Management Accounting

The term management accounting refers to accounting for the management. The management accounting provides information to the management so that planning, organising directing and controlling of business operations can be done in an orderly manner.

Scope of Management Accounting

Following areas are identified within management accounting

1 Financial accounting
2 Cost accounting
3 Revaluation accounting
4 Budgetary control
5 Inventory control
6 Statistical methods
7 Interim Reporting
8 Taxation
9 Office services
10 Internal Audit

Accounting terms

Business transaction

A business transaction is "The movement of money and money's worth form one person to another" or exchange of values between two parties.

Purchase means goods purchased by a businessman from suppliers.

Sales is goods sold by a businessman to his customers.

Purchase Return or Rejection in or Outward Invoice

Purchase return means the return of the full or a part of goods purchased by the businessman to his suppliers.

Sales Return or Rejection out or Inward Invoice

Sales return means the return of the full or a part of the goods sold by the customer to the businessman.

Assets

Assets are the things and properties possessed by a businessman in business.

Two types of Assets

Fixed Asset means the asset remain in business of use and not for resale. Eg. A shop owner purchase buildings, typewriter, showcases

Current Asset means the things and properties for resale ie. The asset converts into cash. Eg. A cloth shop owner buys cloth for resale. Stock of cloth is current asset.

Liabilities

All the amounts payable by a business concern to outsiders are called liabilities.

Capital

Capital is the amount invested for starting a business by a person.

Debtor

Debtor is the person who receives benefit without giving money or moneys worth immediately, but liable to pay in future. i.e. the person owes amounts to the businessman.

Creditor

Creditor is the person who gives benefit without receiving money or money's worth immediately but ot claim in future. i.e. the personto whom amounts are owed by the businessman.

Debit: The receiving aspect of a transaction is called debit or Dr.

Credit: The giving aspect of a transaction is called credit or Cr.

Drawings

Drawings are the amounts withdrawn (taken back) by the businessman from his business for his personal, private and domestic purpose. Drawings may be made in the form cash, goods and assets of the business.

Receipts

It is a document issued by the receiver of cash to the giver of cash acknowledging the cash received voucher.

Account

Account is a summarized record of all the transactions relating to every person, everything or property and every type of service.

Ledger

Ledger is the main book of account. It is the book of final entry where accounts lie.

Journal

Journal is a book of first entry. Transactions are entered in the journal before taken to the appropriate ledger accounts.

Trial Balance

It is a statement of all the ledger account balances prepared at the end of particular period to verify the accuracy of the entries made in books of accounts.

Profit

Excess of credit side total amount over debit side total amount.

Loss

Excess of debit side total amount over credit side total amount.

Profit and loss account

It is prepared to ascertain actual profit or loss of the business.

Balance Sheet

To ascertain the financial position of the business. It is a statement of assets and liabilities.

Types of accounts

Accounts can be divided into two major types.

Personal Accounts

Impersonal Accounts

Impersonal Accounts can be further divided into two types.

1 Real Account

2 Nominal Account

Personal accounts are the accounts of persons, firms, concerns and institutions which the businessmen deal.

Principles: Debit the receiver, Credit the giver

Real Account: These are the accounts of things, materials, assets & properties. It has physical existence which can be seen & touch. Ex. Cash, Sale, Purchase, Furniture, Investment etc.

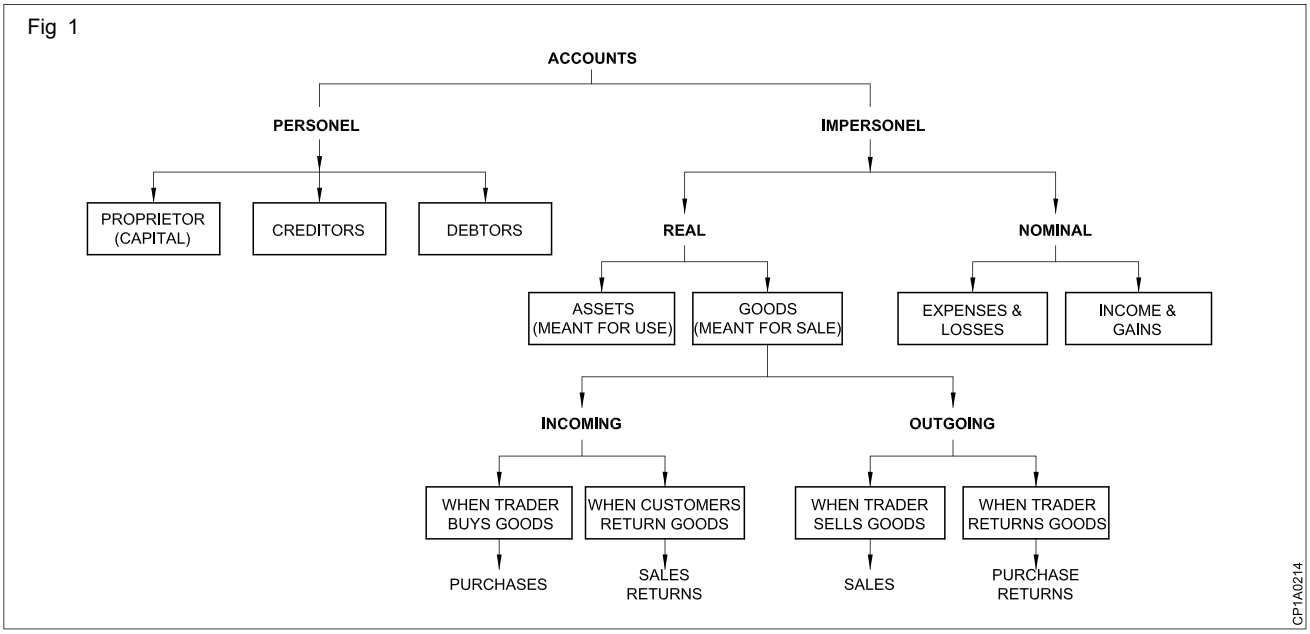
Principles: Debit what comes in, Credit what goes out

Nominal account: Nominal account is the account of services received (expenses and Losses) and services given (income and gain) Ex. Salary, Rent, Wages, Stationery etc.

Principles: Debit all expense/losses, Credit all income/gains

We can clearly understand the classification of accounts in the following Figure 1.

Fig 1



GOLDEN RULES OF ACCOUNTING

Account type	Rules	
Personal Accounts	Debit the Receiver	Credit the Giver
Real Accounts	Debit what comes in	Credit what goes out
Nominal Accounts	Debit all expenses & losses	Credit all incomes & gains

Account Layout

An account has two sides. The left hand side is known as 'Dr' or 'Debit' side. The right hand side is known as 'Cr' or 'Credit' side.

The benefits received by the account are recorded on the left hand side. The benefits imparted by the account are recorded on the right hand side.

The layout of an account looks like as under.

ACCOUNT

Dr Cr

Date	Particulars	Amount	Date	Particulars	Amount
Benefits received			Benefits Imparted		

The receiving aspect which is known as 'Debit' is entered on the Debit side of the account. The giving aspect which is known as 'Credit' is entered on the Credit side of the accounts.

The principle under which both Debit and Credit aspects are recorded is known as the principle of Double entry. Every debit must have a credit and vice versa. If the accounts are not maintained under double entry system, then the records are incomplete and known as single entry.

Double Entry : A business transaction is a transfer of money or money's worth from one account to another. A transfer requires two accounts. A business transaction affects two account's in the opposite directions. If one account receives a benefit, the another account should impart the benefit.

The principle of Double entry is based on the fact that,

There is no giving without receiving and

There is no receiving without giving

DOUBLE ENTRY VS SINGLE ENTRY

Sl. No.	Double Entry	Single Entry
1	For every Debit there is a corresponding credit and vice versa	There are no credits and Debits here
2	Maintains a complete record of a Personal accounts b Real accounts and c Nominal accounts	An incomplete record. Only personal accounts and cash accounts are maintained
3	A balance sheet and profit and loss statement can be prepared conveniently, since the books of accounts present a complete picture	A balance sheet and profit and loss statement cannot be conveniently prepared since the accounting records are incomplete
4	Double Entry is a complete, scientific system of keeping books of accounts	Single Entry is not a system. It is incomplete and unscientific

Personal Account - Examples:

1 Sold goods to Selvan on credit Rs.1,100/-

Selvan account receives a benefit and hence should be debited

2 Returned damaged goods to Sami.

Sami account receives a benefit and hence should be debited

3 Proprietor Thiru Anbu withdraws cash Rs.500 for house hold expenses

Anbu - Drawing account receives a benefit and hence should be debited

In the above examples selvan a/c, Sami a/c, and Anbu - Drawings a/c are Personal accounts. They are receivers of benefits and hence should be debited.

4 Anbu started business with cash Rs.50,000/-

Anbu - Capital a/c gives benefit in the form of cash to business. Hence capital a/c should be credited

5 Bought goods from Somu on credit for Rs.1,700/-

Somu a/c gives a benefit and hence should be credited

6 Received five chairs from Godrej Co. at Rs.45 per chair on credit basis

Godrej Co a/c gives benefit in the form of 5 chairs. Hence godrej co a/c should be credited.

In the above examples capital a/c, Somu a/c and Godrej a/c are personal accounts. They are giving benefits. Hence their accounts should be credited.

Real Account - Examples

1 Bought five chairs [furniture] from Godrej Co. at Rs.45 per chair on credit basis.

Furniture worth Rs.225 have come into the business as per Rule, Debit what comes in. Since Furniture has come in, Furniture Account should be debited.

2 Anbu started business with cash Rs.50,000/-

Cash of Rs.50,000 has come into the business and hence cash account should be debited

3 Purchased goods from Somu on credit for Rs.1,700/-.

Goods are bought in the aspect of purchases. Hence purchases account should be debited.

In the above examples, Furniture Account, cash account and purchases account are real accounts. Since they have come into business, the accounts are debited.

4 Sale of goods to Selvan on credit Rs.1,100/-

Goods have gone out of the business hence sales account is credited

5 Paid cash to Somu Rs.1,700/-

Cash goes out of the business and hence cash account should be credited

6 Paid rent Rs.250/-

Cash goes out of the business and cash should be credited

In the above examples Sales Account Cash Accounts are real accounts. They are credited as per Rule.

Nominal Accounts Examples:

1 Paid Rent Rs.250/-

Rent is an expense account. Hence rent account should be debited

2 Paid salary Rs.1,200/-

Salary is an expense account. Hence salary account should be debited

3 Purchase of paper, pencils, ink, cover's etc., for Rs.250/-

These are stationary items and expense items. Hence stationery account should be debited

In the above three examples Rent account, salary account and stationery account all are nominal accounts. They are debited since they are expense items

4 Received commission Rs.500/-

Here commission is income to the business and hence commission account should be credited.

5 Received interest on loan given to B Rs.100/-

Interest is income to the business. Hence interest Account should be credited.

In the above two examples commission account and interest account are Nominal accounts. They have been credited since they are incomes.

Books of Accounts

Books of accounts can be generally classified into three categories.

1 Journal

2 Ledger

3 Subsidiary Books

Journal : Journal is a book of prime entry. When a transaction takes place, it is recorded in the Journal.

Layout of a Journal

Date (1)	Particulars (2)	L.F. (3)	Dr. (4)	Cr. (5)

L.F means Ledger Folio. The L.F column is meant for recording the page number of the concerned account in the Ledger. The transactions recorded in the journal will be posted to the appropriate accounts in the Ledger.

Journalising : Journalising is the process of analysing the business transaction under the heads of debit and credit and recording them in the journal.

When journalising a transaction the following steps to be followed.

1 What are the accounts affected? name them.

2 What type of accounts are they? Identify them. e.g., Personal, Real or Nominal

3 Apply the rules for debit and credit.

Account type	Debit	Credit
Personnal Accounts	The Receiver	The Giver
Real Accounts	What comes in	What goes out
Nominal Accounts	Expenses and Losses	Incomes and Gains

In Journalising a transcation, the debit aspect is shown first with observation "Dr" after the name of the account. The credit aspect is shown as a second item with the word "To" at the beginning. A brief description of the transaction known as "Narration" is given at the end of every entry in the journal.

Example 1

1st July 2003 : Received cash from Muthu Rs.500

1 What are the accounts affected?

Cash Account and Muthu Account

2 Types of Accounts

Cash account is a Real account

Muthu account is a personal account

3 According to Rule No.2 Cash Account should be debited because cash comes into the business.

According to Rule No.1, Muthu Account should be credited since Muthu has given benefit.

Journal

Date	Particulars	L.F.	Dr.		Cr.	
			Rs.	P.	Rs.	P.
2003 July 1st	Cash Account Dr. To Muthu Account (Being cash received from Muthu)	2 41	500	00	500	00

Example 2

July 7, 2003 : Bought goods for cash Rs.1543.

1 Purchase Account and Cash Account are affected.

2 Purchase Account and cash accounts are Real accounts.

3 Purchase Account should be debited since goods have come into business. Cash account should be credited since cash has gone out.

Date	Particulars	L.F.	Dr.		Cr.	
			Rs.	P.	Rs.	P.
2003 July 7th	Purchase Account Dr. To cash Accounts (Being cash purchases of goods)	35 4	1543	00	1543	00

Ledger

The ledger is the main book of business containing Personal, Real and Nominal accounts of the business. But transactions are not recorded in the Ledger directly. These are first entered in the Journal and then posted to the concerned accounts in the Ledger.

Layout of Ledger account

An account is divided in the middle and the two sides are called debit side and credit side respectively.

Kannan Account (Personal Account)

Dr.	Cr.
Debit Kannan when he receives goods, money or value from the business	Credit Kannan when he gives goods, money or value to the business

Furniture Account (Real Account)

Dr.	Cr.
Debit Purchase of Asset	Credit sale of asset

Salary Account (Nominal Account)

Dr.	Cr.
Debit expenses in normal account	

Interest received account (Nominal a/c)

Dr.	Cr.
	Credit gains in Nominal accounts

Posting in the Ledger : The technique of recording the journal entries into the ledger is known as posting.

The account end with "Dr" is to be debited in the Ledger. The account starts with "To" is to be credited in the Ledger. It is customary to write "To" on the debit side and "By" on the credit side.

Example:

Given the journal Entry

Date	Particulars	L.F.	Dr.		Cr.	
			Rs.	p.	Rs.	p.
2003 July 1st	Cash Account Dr. To Capital Account (Being the amount invested in business)		75,000	00	75,000	00

Capital account starts with "To". So the capital account is credited with the entry in the credit side as "By cash".

CAPITAL ACCOUNT

Dr. Cr.

Date	Particulars	Amount Rs. P.	Date	Particulars	Amount	
					Rs.	P.
			2003 July 1	By cash	75,000	00

JOURNAL VS LEDGER

S.No.	Journal	Ledger
1	The journal is a subsidiary book	The ledger is the main book of accounts
2	Transactions are first entered in Journal	Entries in the journal are posted in the Ledger.
3	Journal is a daily record. Business transactions are entered in this book in the order of dates	Posting from journal to Ledger is done periodically, may be weekly or fortnightly.
4	Entering the transaction in the journal is called journalising	The act of recording journal entries into ledger is called "Posting".

Subsidiary Books : Journal is sub-divided into smaller journals called subsidiary-journals. We can synonymously call subsidiary - journals as subsidiary books.

Categories of subsidiary book includes the following.

1 Purchase Book.

2 Sales Book.

3 Purchase Returns Books

4 Sales Returns Books

5 Cash book

1 Purchase Book

– This book is also called as "Purchase Day book", "Invoice Book", "Bought Book", "Purchases Journal".

– It is used for recording credit purchases of goods.

– Cash purchases are not recorded in this book.

Layout of purchase book

Date	Name of the Supplier	L.F.	Invoice No.	Amount

2 Sales Book

This book is also called as "Sales Journal", "Sold Book", "Sales Day Book"

– It is used to record credit sales of goods.

– Cash sales are not recorded in this sales book

– Sale of an old asset on credit is not recorded in the sales book

Layout of the Sales Book

Date	Name of the Customer	L.F.	Outward Invoice No.	Amount

3 Purchase Returns Book

– This book is also called as "Purchases Returns Journal", "Returns Outwards Book".

– It is used to record transactions relating to return of goods to suppliers.

Layout of Purchase Returns Book

Date	Name of Supplier	L.F.	Debit Notes Nos.	Amount

4 Sales Returns Book

– This book is also called as "Returns Inwards Book", "Sales Returns Journal".

– It is used to record particulars of goods returned by customers.

Layout of Sales Returns Book

Date	Name of Customer	L.F	Credit Note No.	Amount

5 Cash Book

– The purpose of the cash book is to record all cash receipts and payments and the sums deposited into and with drawn from the bank.

– Cash book looks just like a ledger.

Layout of Cash Book

Dr Cr

Date	Particulars	L.F.	Amount	Date	Particulars	L.F	Amount

Accounts Information - Grouping

Current Asset: It is converted into cash with in a year. Ex. Bills receivable.

Direct Expenses: These are the expenses which are directly related to manufacturing of goods. Ex. Wages, factory rent, heating, lighting etc.,

Indirect Expenses: These are the expenses which are indirectly related to manufacturing of goods. Ex. Salary, rent, stationery, advertisement, printing.

Depreciation: Decrease the value of the asset.

Sundry debtors: The person who is the receiver or customer.

Sundry creditors: The person who gives or supplier.

Expenses Outstanding or Unpaid expenses or Expenses due:

Expenditure incurred during current year but the amount on which is not yet paid. (Added to the expenditure on the debit side and entered on the liability side.)

Income received in advance or Income received but not earned.

Income received during the currentyear but not earned or a part of which relates to the next year. (Deducted from the concerned income on the credit side and entered on the liability side)

Prepaid advance or Expenses

Expenditure paid during current year but not incurred or a part of which related to the next year is called expenditure prepaid.

Income outstanding means income earned during the current year but the amount on which is not yet received.

Accounts information - Ledger - Grouping

Ledger	Group
Cash at bank	Bank Account
Bank overdraft	Bank OD
Capital	Capital Account
Cash in hand	Cash in hand
Other Liabilities	Current Liabilities
Bills receivable	Current Asset
Stock of stationery	Current Asset
Prepaid expenses	Current Asset
Income outstanding	Current Asset
Bills payable	Current Liabilities
Expense outstanding	Current Liabilities
Income received in advance	Current Liabilities
Fixed deposit at bank	Deposit
Fright charges	Direct Expenses
Carriage inwards or Purchases	Direct Expenses
Cartage and coolie	Direct Expenses
Octroi	Direct Expenses
Manufacturing wages	Direct Expenses
Coal, gas, water	Direct Expenses
Factory rent, insurance, electricity, lighting and heating	Direct Expenses
Loose tools	Fixed Asset
Fixtures and fittings	Fixed Asset
Furniture	Fixed Asset
Motor Vehicles	Fixed Asset
Plant and machinery	Fixed Asset
Land and building	Fixed Asset
Leasehold property	Fixed Asset
Patents	Fixed Asset
Goodwill	Fixed Asset
Salary	Indirect Expenses
Postage and Telegrams	Indirect Expenses
Telephone charges	Indirect Expenses
Rend paid	Indirect Expenses

	Rates and taxes	Indirect Expenses
	Insurance	Indirect Expenses
	Audit fees	Indirect Expenses
	Interest on bank loan	Indirect Expenses
	Interest on loans paid	Indirect Expenses
	Bank charges	Indirect Expenses
	Legal charges	Indirect Expenses
	Printing and stationery	Indirect Expenses
	General expenses	Indirect Expenses
	Sundry expenses	Indirect Expenses
	Discount allowed	Indirect Expenses
	Carriage outwards or sales	Indirect Expenses
	Travelling Expenses	Indirect Expenses
	Advertisement	Indirect Expenses

Shortcut Keys

F1	Select Company
Alt+F1	Shut Company
Alt+F3	Company information menu
Ctrl+A	To accept a form wherever you use the key combination the screen or report will be accepted as it is on this screen.
Alt+C	To create a master at a voucher screen.
Alt+D	To delete a voucher/To delete a master.
Ctrl+Enter	To alter a master while making an entry or viewing report.
F2	Date
Alt+F2	Change period
F11	Company features
F12	Configuration optionsare applicable to all the companies in a data directory.
Ctrl+N	Calculator screen.
Ctrl+V	Voucher mode (Cr. Dr)/Invoice mode (name of item, rate, quantity, and amount)
F4 Contra	Records funds transfer between cash and bank accounts
F5 Payment	Record all bank and cash payments

F6 Receipt	Records all receipts into bank or cash accounts.
F7 Journal	Records adjustments between ledger accounts
F8 Sales	Records all sales.
Ctrl+F8	Credit note voucher
F9	Purchase Records all purchase.
Ctrl+F9	Debit note voucher
F10	Reversing Journal voucher
Alt+2	To duplicate a voucher
Alt+C	To create a Master at a voucher screen
Alt+E	To export the report in ASCII, SDF, HTML or XML Format
Alt+I	To insert a voucher
Alt+R	To remove a line in a report in reports screen
Alt+S	To bring back a line, you removed using Alt+R in reports screen
Alt+X	To cancel a voucher in Daybook/List of vouchers

Journalize the following transactions

1 Commenced business with cash Rs.50,000/-

2 Deposit into bank Rs.15,000/-

3 Bought office furniture Rs.3,000/-

4 Sold goods for cash Rs.2,500/-

5 Purchased goods form Mr. X on credit Rs.2,000/-

6 Sold goods to Mr. Y on credit Rs.3,000/-

7 Received cash form Mr. Y on account Rs.2,000/-

8 Paid cash to Mr. X Rs.1,000/-

9 Received commission Rs.50/-

10 Received interest on bank deposit Rs.100/-

11 Paid into bank Rs.1,000/-

12 Paid for advertisement Rs.500/-

13 Purchased goods for cash Rs.1,500/-

14 Sold goods for cash Rs.1,500/-

15 Paid salary Rs.5,000/-

Gateway of Tally - Account info - Ledger - Create

Gateway of Tally - Account voucher

Sl.No.	Key	Voucher	Ledger	Group	Types of account	Principles	Amount
1	F6	Receipt	Cr. Capital	Capital account	Personal	Giver	50,000
			Dr. Cash	Cash in hand	Real	Comes in	50,000
2	F4	Contra	Cr. Cash	Cash in hand	Real	Goes out	15,000
			Dr. Bank	Bank account	Real	Comes in	15,000
3	F5	Payment	Dr. Furniture	Fixed asset	Real	Comes in	3,000
			Cr. Cash	Cash in hand	Real	Goes out	3,000
4	F8	Sales	Dr. Cash	Cash in hand	Real	Comes in	2,500
			Cr. Sales	Sales account	Real	Goes out	2,500
5	F9	Purchase	Cr. X	Sundry creditor	Personal	Giver	2,000
			Dr. Purchase	Purchase account	Real	Comes in	2,000
6	F8	Sales	Dr. Y	Sundry debtors	Personal	Receiver	3,000
			Cr. Sales	Sales account	Real	Goes out	3,000
7	F6	Receipt	Cr. Y			Giver	2,000
			Dr. Cash	Cash in hand	Real	Comes in	2,000
8	F5	Payment	Dr. X			Reciever	1,000
			Cr. Cash	Cash in hand	Real	Goes out	1,000
9	F6	Receipt	Cr. Commission	Indirect income	Nominal	Credit all income	50
			Dr. Cash	Cash in hand	Real	Comes in	50
10	F6	Receipt	Cr. Interest on bank deposit	Indirect income	Nominal	Credit all income	100
			Dr. Bank	Bank account	Real	Comes in	100
11	F4	Contra	Cr. Cash	Cash in hand	Real	Goes out	1,000
			Dr. Bank	Bank account	Real	Comes in	1,000
12	F5	Payment	Dr. Advertisement	Indirect expenses	Nominal	Debit all expenses	500
			Cr. Cash	Cash in hand	Real	Goes out	500
13	F9	Purchase	Cr. Cash	Cash in hand	Real	Goes out	800
			Dr. Purchase Cr. Cash	Purchase account	Real	Comes in	800
14	F8	Sales	Dr. Cash	Cash in hand	Real	Comes in	1,500
			Cr. Sales	Sales account	Real	Goes out	1,500
15	F5	Payment	Dr. Salary	Indirect expense	Nominal	Debit all expenses	5,000
			Cr. Cash	Cash in hand	Real	Goes out	5,000

Financial Accounting Reports

Objectives: At the end of this lesson you shall be able to
- **understand financial accounting**
- **understand book-keeping**
- **prepare trading account, profit and loss account & balance sheet.**

Introduction: The accounting for revenues, expenses, assets and liabilities that is commonly carried on in the general offices of a business is termed as Financial Accounting.

Its aim is to ascertain the profit or loss of the business and states the financial position of the business as at a particular date.

Financial accounting includes the following activities.

i Book keeping

ii Preparation of Trading Account

iii Preparation of profit and loss account

iv Preparation of Balance Sheet

Book-keeping : Book-keeping is the art of recording business transactions in a systematic manner. Main objective of book-keeping is to calculate the profit or loss of a business accurately.

We have discussed various types of Books of accounts in the previous chapter.

Advantages of Book-Keeping

1 Book-keeping provides reliable record of transactions essential for ready reference.

2 Profit or loss is ascertained using Books of accounts

3 Calculation of due amount the businessman has to pay others is done using Books of accounts

4 Borrowings and Assets are controlled.

5 Financial position and growth of business is ascertained

6 Do's and Don'ts are identified

7 Book-keeping is used for taxation and fixing selling price.

Final Account : Trading, Profit and Loss account, Balance sheet are prepared at the end of the year or at the end of the part. So it is called Final Account.

– Trading, Profit & loss account is prepared to find out profit or loss of the organisation

– Balance sheet is prepared to find out the financial position of the organisation

Trading Account : Trading refers to buying and selling of goods. Trading account shows the result of buying and selling of goods. This account is prepared to find out the difference between the selling price and cost price.

– If the selling price exceeds the cost price, it will bring Gross Profit. For example, If the goods of cost price Rs.50,000 are sold for Rs.60,000 that will bring in Gross Profit of Rs.10,000.

– If the cost price exceeds the selling price, it will bring Gross loss. For example, if the goods of cost price Rs.60,000 are sold for Rs.50,000, that will result in Gross Loss of Rs.10,000

Thus Gross Profit or Gross Loss is indicated in Trading Account.

Items appearing in the Debit side of Trading account

1 **Opening Stock :** Stock on hand at the commencement of the year or period is termed as the opening stock

2 **Purchases :** It indicates total purchases both cash and credit made during the year

3 **Purchases returns or Returns outwords :** Purchases Returns must be subtracted from the total purchases to get the net purchases. Net purchases will be shown in the trading account.

4 **Direct Expenses on Purchases :** Some of the Direct Expenses are

   a **Wages:** It is also known as productive wages or Manufacturing wages

   b **Carriage or carriage Inwards:**

   c **Octroi Duty :** Duty paid on goods for bringing them within municipal limits

   d Customs duty, Dock duty, clearing charges, Import duty etc.,

   e Fuel, Power, Lighting charges related to production

   f Oil, Grease and Waste

   g **Packing charges:** Such expenses are incurred with a view to put the goods in a saleable condition.

Items appearing on the credit side of trading account

1 **Sales:** Total sales (Including both - cash and credit) made during the year or period

2 **Sales Returns or Return Inwards:** Sales returns must be subtracted from the Total sales to get Net sales. Net sales will be shown

3 **Closing stock:** Generally, closing stock does not appear in the Trial Balance. It appears outside the Trial Balance. It represents the value of goods at the end of the trading period.

Specimen form of a trading a/c

Dr. **Trading Account for the year ending** **Cr.**

Particulars	Amount Rs. P.	Amount Rs. P.	Particulars	Amount Rs. P.	Amount Rs. P.
To Opening stock		xxxx	By sales	xxxx	
To Puchase	xxxx		Less: Returns		
Less : Returns-outwards	xxxx		Inwards	xxxx	xxxx
To wages		xxxx	By closing stock		xxxx
To Freight		xxxx	By Gross loss		xxxx
To Carriage Inwards		xxxx	(to be transferred to P&L A/c)		
To Clearing charges		xxxx			
To Packing charges		xxxx			
To Dock dues		xxxx			
To Power		xxxx			
To Gross Profit (to be transferred to P&L A/c)		xxxx			
		xxxx			xxxx

Balancing of Trading Account : The difference between the two sides of the Trading account, indicates either Gross Profit or Gross loss. If the total on the Credit side is more, the difference represents Gross Profit. On the other hand, if the total of the debit side is high, the difference represents Gross loss. The Gross Profit or Gross Loss is transferred to Profit & Loss A/c.

Closing Entries of Trading A/c : Trading account is a ledger account. Hence no direct entries should be made in the trading account. Several items such as purchases, sales are first recorded in the journal and then posted to the ledger. The same accounts are closed by transferring them to trading account. Hence it is called as closing entries.

Advantages of Trading Account

1 The results of purchases and sales can be clearly assertained.

2 Gross profit ratio to sales could also be easily ascertained. It helps to determine price

3 Gross profit ratio to direct expenses could also be easily ascertained. And so, unnecessary expenses could be eliminated

4 Comparison of trading account details with previous year details help to draw better administrative policies.

Example 1

Prepare a trading account from the following informations of a trader

i Total purchases made during the year 2002 Rs.20,000

ii Total sales made during the year 2002 Rs.25,000

Trading account for the year ending 31st December 2002

Dr Cr.

Particulars	Amount Rs. P.	Particulars	Amount Rs. P.
To purchases	20,000	By sales	25,000
To Gross profit c/d	5,000		
	25,000		25,000

Example 2

Prepare a Trading Account from the following informations of a trader.

i 2002 Jan 1 Opening stock Rs.5,000

ii Purchases Rs.16,100

iii Sales Rs.30,600

iv 2002 Dec 31 Closing Stock Rs.3,500

Trading Account for the year ending 31-12-2002

Dr Cr.

Particulars	Amount Rs. P.	Particulars	Amount Rs. P.
To opening stock	5,000	By sales	30,600
To purchases	16,100	By Closing stock	3,500
To Gross profit c/d (Transferred to P&L a/c)	13,000		
	34,100		34,100

PROFIT and LOSS Account : Trading account reveals Gross profit or Gross loss. Gross profit is transferred to credit side of profit and loss account. Gross loss is transferred to debit side of the profit and loss account. Thus profit and loss account is commenced. This profit & loss account reveals Net Profit or Net loss at a given time of accounting year.

Items appearing on Debit side of the P& L Account:

The expenses incurred in a business is divided into two parts. One is Direct expenses which are recorded in the trading account. Another one is indirect expenses which are recorded on the debit side of profit & loss account.

Indirect Expenses are grouped under four heads:

1 **Selling Expenses :** All expenses relating to sales such as carriage outwards, travelling expenses, advertising etc.,

2 **Office Expenses :** Expenses incurred on running an office such as office salaries, rent, tax, postage, stationery etc.

3 **Maintenance Expenses :** Maintenance expenses of assets. It includes repairs and renewals, depreciation etc.,

4 **Financial Expenses :** Interest paid on loan, discount allowed etc., are few examples for Financial expenses.

Items appearing on credit side of P & L account : Gross profit is appeared on the credit side of Profit and Loss account. Also other gains and incomes of the business are shown on the credit side. Typical of such gains are items such as Interest received, Rent received, Discounts earned, Commission earned.

Specimen Form:

Profit & Loss Account for the year ended 31st March 2002

Dr Cr.

Particulars	Amount Rs. P.	Particulars	Amount Rs. P.
To Trading a/c (Gross loss)	xxxx	By Trading a/c (Gross profit)	xxxx
To Salaries	xxxx	By Commission earned	xxxx
To Rent & Taxes	xxxx	By Rent received	xxxx
To Stationaries	xxxx	By Interest received	xxxx
To Postage expenses	xxxx	By Discounts received	xxxx
To Insurance	xxxx	By Net loss	xxxx
To Repairs	xxxx	(Capital account)	
To Trading expenses	xxxx		
To Office expenses	xxxx		
To Interest	xxxx		
To Bank charges	xxxx		
To Establishment expenses	xxxx		
To Sundry expenses	xxxx		
To Commission	xxxx		
To Discount	xxxx		
To Advertisement	xxxx		
To Carriage Outwards	xxxx		
To Travelling expenses	xxxx		
To Distribution expenses	xxxx		
To Bad debts	xxxx		
To Depreciation	xxxx		
To Net profit (transferred to Capital a/c)	xxxx		
	xxxx		xxxx

Example 3

Prepare profit and loss account from the following balances of Dharani Enterprises for the year ending 31.12.2012

i Office rent Rs.3,000 Salaries Rs.8,000

ii Printing expenses Rs.2,200 Sationeries Rs.2,400

iii Tax, Insurance Rs.1,400 Discount allowed Rs.600

iv Discount received Rs.400

v Travelling expenses Rs.2,600

vi Advertisement Rs.3,600

vii Gross profit transferred from the trading account Rs.25,000

Profit and Loss Account of Dharani Enterprises for the year ending 31st Dec 2012

Dr Cr.

Particulars	Amount Rs. P.	Particulars	Amount Rs. P.
To Salaries	8,000	By Gross profit	25,000
To Office rent	3,000	(transferred from Trading a/c)	
To Stationaries	2,400	By Discount received	400
To Printing expenses	2,200		
To Tax, insurance	1,400		
To Discount allowed	600		
To Travelling expenses	2,600		
To Advertisement	3,600		
To Net profit	1,600		
(Capital account)			
	25,400		25,400

Example 4

Prepare trading and Profit - Loss account for the year ending 31st March 2012 from the books of Swamy & Co.,

	Rs.
Stock (15-01-1994)	15,000
Carriage Outwards	4,000
Purchases	1,65,000
wages	10,000
Purchase returns	10,000
Sales Returns	5,000
Postage	300
Salaries	5,000
Discount received	500
Stationaries	1,000
Bad debts	100
Interest	800
Sales	3,00,000
Insurance	400
Closing stock	80,000

Trading and Profit and Loss account of Swamy & Co., for the year ending 31st March 2012

Dr Cr.

Particulars	Amount Rs. P.	Particulars	Amount Rs. P.
To Stock (15.01.1994)	15,000	By sales 3,00,000	
To Purchases 1,65,000		Less returns 5,000	2,95,000
Less Returns 10,000	1,55,000	By closing stock	80,000
To Wages	10,000		
To Gross Profit	1,95,000		
	3,75,000		3,75,000
To Salaries	5,000	By Gross profit	1,95,000
To Postage	300	(transferred from trading a/c)	
To Bad depts	100	By discount received	500
To Carriage outwards	4,000		
To Stationaries	1,000		
To Interest	800		
To Insurance	400		
To Net profit	1,83,900		
(Capital a/c)	1,95,500		1,95,500

If trial balance shows trading expenses as well as office expenses, the Trading expenses should be shown in the trading account and office expenses should be shown in profit and loss account. On the otherhand if the trial balance shown only a trading expenses, it should be shown in the profit and loss account

In the trial balance, wages are clubbed with salaries as 'wages and salaries'. This item is shown in Trading account. On the other hand, it appears as salaries and wages and this item is shown in the profit & Loss account.

Income tax : Income tax paid by a proprietor is considered as personal expenses. So instead of debited to Profit and Loss account, it should be debited to the capital account.

Balance sheet

– Trading account provides the details of Gross Profit or Gross Loss

– Profit and Loss account provides the details of the Net Profit or Net Loss

– Besides the above, the proprietor wants

i to know the total assets invested in business

ii To know the position of owner's equity

iii To know the liabilities of business

Definition : The word "Balance Sheet' is defined as " a statement which sets out the assets and liabilities of a business firm and which serves to ascertain the financial position of the same on any particular date".

– On the left hand side of this statement, the liabilities and capital are shown

– On the right hand side of this statement, all the assets are shown

– Hence both the sides of the balance sheet must be equal

– Capital arrives assets exceeds the liabilities

Objectives of Balance sheet

1 It shows accurate financial position of a firm

2 It shows various transactions took place at a given period

3 It indicates that, whether the firm has sufficient assets to repay in liabilities

4 The accuracy of final accounts is verified by this statement

5 It shows the profit and loss arrived, through profit & Loss account

SPECIMEN FORM OF A BALANCE SHEET:
BALANCE SHEET OF SUNIL ENTERPRISES AS AT 31.12.2012

Dr Cr.

Liabilities	Amount Rs. p.	Amount Rs. P.	Assets	Amount Rs. P.	Amount Rs. P
Sundry creditors		xxxx	Cash in hand		xxxx
Bills payable		xxxx	Cash at Bank		xxxx
Bank overdraft		xxxx	Bills receivable		xxxx
Loans		xxxx	Sundry Debtors		xxxx
Mortgage		xxxx	Closing stock		xxxx
Reserve Fund		xxxx	Furniture & Fittings		xxxx
Outstanding exp.		xxxx	Investments		xxxx
Capital	xxxx		Plant & Machinery		xxxx
Add: Net profit (or)			Loose tools		xxxx
Less: Net loss	xxxx		Land and Buildings		xxxx
	xxxx		Business premises		xxxx
Less: Drawings	xxxx		Horses & carts		xxxx
	xxxx		Prepaid exp.		
Less : Income tax	xxxx	xxxx	Patents & Trade marks		xxxx
			Good will		xxxx
		xxxx			xxxx

The Balance sheet contains two parts.

1 Left hand side the liabilities

2 Right hand side the assets

Assets : Assets represent everything which a business owns and has money value. Assets are always shown as debit balance in the ledger. Assets are classified as follows:

1 **Tangible Assets :** Assets which can be seen and felt by touching are called Tangible assets. Tangible assets are classified into two:

a **Fixed Assets :** Assets which are durable in nature and used in business over and again are known as Fixed assets. E.g., Land and building, machinery, trucks, etc.,

b **Floating Assets or Current Assets :** Current Assets are

i Meant to be converted into cash

ii Meant for resale

iii Likely to undergo change

E.g: Cash, Bank Balance, stock, sundry debtor

2 **Intangible Assets :** Assets which cannot be seen and has no fixed shape. E.g., Good will, Patent

3 **Fictitious Assets :** Assets which have no real value and will appear on the assets side of balance sheet are known as fictitious assets.

Ex: Preliminary expenses, Discount on creditors

Liabilities : All that the business owes to others are called liabilities. It means that any amount which a business concern has to pay legally. It also includes proprietor's capital. They are known as credit balances in ledger.

Liabilities are classified as follows.

1 **Long term liabilities :** Liabilities which will be redeemed after a long period of time say 10 to 15 years. e.g., capital, Long-term Loans

2 **Current Liabilities :** Liabilities which are redeemed within a year are called current liabilities or short term liabilities e.g., Trade creditors, Bank Loan

3 **Contingent Liabilities :** Liabilities which have the following features are called contingent liabilities. They are :-

a Not actual liability at present

b Might become a liability in future on condition that the contemplatted event occurs. e.g., Liability in respect of pending suit

Equation of Balance sheet		
Capital	=	Assets - Liabilities
Liabilities	=	Assets - Capital
Assets	=	Liabilities + Capital

Trial Balance : When the transactions are recorded under double entry system, there is a credit for every debit. When one account is debited, another account is credited with equal amount.

If a statement is prepared with debit balances on one side and credit balances on the other side, the totals of the two sides will be equal such a statement is called Trial Balance.

Advantages:

1 It is the shortest method of verifying the arithmetical accuracy of entries made in the ledger. If the Trial balance agrees, it is an indication that the accounts are correctly written up. But it is not a conclusive proof.

2 It helps to prepare the trading account, Profit and loss account and balance sheet

3 It presents to the business man a consolidated list of all Ledger Balances.

Sundry Debtors : There may be large number of deptors to a trader. All the deptor's names are not written in the trial balance. A list of debtors with their individual debit balances are prepared and totalled. The total is written under the heading "Sundry Debtors" which appears in the Trial Balance.

Sundry Creditors : There are a number of parties from whom the Trader buys goods on credit basis. All these creditors names are not written in the Trial Balance. A list of creditors with the balances due to them is prepared and totalled. The total is written under the heading "sundry creditors" which appears in the Trial Balance.

Preparation of Balance sheet : Once the Trial Balance is arrived, using that Trading account, Profit and Loss account and Balance Sheet can be casted.

Trial Balance Vs Balance sheet

Trial Balance	Balance Sheet
1 It shows the balances of all ledger accounts	It shows the balances of personal and real accounts only
2 It is prepared after the completion of the ledger accounts or arrival of the balances	It is prepared after the completion of trading and profit and loss account
3 Its object is to check the arithmetical accuracy	Its object is to reveal the financial position of the business
4 Items shown in the trial balance are not in order	Items shown in Balance sheet must be in order
5 It shows Opening Stock	It shows Closing Stock
6 It has the headings, debit and credit.	It has the headings of Assets and Liabilities

Example 5

From the follwing trial balance extracted from the books of Sundar & Sons as on 31.12.2012. Prepare i. Trading and Profit and Loss account and ii. Balance sheet.

Trial Balance as on 31-12-2012

Dr. Cr.

Debit balances	Amount Rs. P.	Credit Balances	Amount Rs. P.
Cash in hand	2,000	Capital	2,00,000
Machinery	60,000	Sales	2,54,800
Stock	50,000	Sundry Creditors	40,000
Bills receivable	1,600	Bank overdraft	22,000
Sundry Debtors	50,000	Return outwards	3,000
Wages	70,000	Discounts received	1,800
Land	40,000	Bills payable	1,800
Carriage inwards	2,400		
Purchases	1,80,000		
Salaries	24,000		
Rent	4,000		
Postage	1,000		
Return inwards	3,200		
Drawings	10,000		
Furniture	18,000		
Interest	600		
Cash at Bank	6,600		
	5,23,400		5,23,400

Stock as on 31-12-2012 is Rs.1,00,000

Trading, Profit & Loss account of Sundar & Sons for the year ending 31-12-2012

Dr Cr.

Particulars	Amount Rs. P.	Particulars	Amount Rs. P.
To stock (1-1-2012)	50,000	By sales 2,54,800	
To purchases 1,80,000		Less Returns 3,200	2,51,600
Less returns 3,000	1,77,000	By closing stock	1,00,000
To wages	70,000		
To carriage inwards	2,400		
To Gross Profit c/d	52,200		
(Transferred to P&L a/c)	3,51,600		3,51,600
To salaries	24,000	By Gross Profit b/d	52,200
To Rent	4,000	(Transferred from Trading a/c)	
To postage	1,000	By discount received	1,800
To interest	600		
To Net Profit (capital a/c)	24,400		
	54,000		54,000

Balance sheet of Sundar & Sons as at 31.12.2012

Dr Cr.

Liabilities	Amount Rs. P.	Assets	Amount Rs. P.
Sundry creditors	40,000	Cash in hand	2,000
Bank overdraft	22,000	Cash at bank	6,600
Bills payable	1,800	Machinery	60,000
Capital 2,00,000		Bills receivable	1,600
Add: Net profit 24,400		Sundry debtors	50,000
Less: Drawings 2,24,400	2,14,400	Furniture	18,000
		Land	40,000
		Closing stock	1,00,000
	2,78,200		2,78,200

Costing Systems

Objectives: At the end of this lesson you shall be able to
- **understand costing, its types and cost classification**
- **differentiate budgeting and standard costing**
- **cost centre, cost category, cost centre reports**
- **inventory accounting with tally**
- **inventory report, inventory books, statement of inventory.**

Introduction : Costing refers to fixing the costs of a product. The factors which determines the cost of a product are known as elements of cost.

Elements of cost : There are three cost elements exist in costing. They are

1 Material cost

2 Labour cost

3 Expenses

- Material cost refers to the cost of raw materials used for production of a product.
- Labour cost refers to the wages paid to the workers in the manufacturing department.
- Expenses refers to the expenditure by the way of rent depreciation and power cut.

Concept of Direct and Indirect costs : The total expenditure may be classified as Direct cost and Indirect cost.

Direct cost : The expenditure which can be conveniently allocated to a particular job or product or unit of service is known as direct cost.

Direct expenditure is made up of

1 Direct materials

2 Direct labour

3 Direct expenses

Indirect cost : The expenditure which cannot be conveniently allocated to a particular job or product or unit of service is known as indirect cost.

In a firm producting a larger variety of articles most of the expenditure apart from materials and labour will be indirect.

Indirect expenditure is made up of

1 Works of factory expenses

2 Office and administrative expenses

3 Selling and distributive expenses

Cost classification : Cost classification is the process of grouping costs according to their common characteristics.

Costs may be classified according to their nature and number of characteristics such as function, variability, controllability and normality.

1 Nature : Costs are classified according to their nature as

a Materials cost

b Labour cost

c Expenses

2 Function : According to the divisions of activity, costs can be classified as

a Production cost

b Admistrative cost

c Selling cost

d Distribution cost

3 Variability : According to their behaviour in relation to changes in the volume of production cost can be classified as

a Fixed cost

b Semi fixed cost

c Variable cost

4 Controllability : Costs are classified according to their influences by the action of a given member of an undertaking as

a Controllable cost

b Uncontrollable cost

5 Normality : Costs are classified according to the costs which are normally incurred at a given level of output as

a Normal cost

b Abnormal Cost

Presentation of total cost : The presentation of total cost according to their nature is shown here.

Statement of total cost

	Rs.
Direct material cost	----------
Direct wage	----------
Direct expenses	----------
"A" Prime cost	----------
Add : Works on cost or	
Factory expenses	----------
"B" works cost	----------
Add: Office and administrative	
Expenses	----------
"C" cost of production	----------
Add: Selling and Distribution	
expenses	----------
"D" Cost of sale	----------
Add: Profit or Less: loss	----------
	_____
"E" Selling price	----------
	_____

The presentation of total cost according to their variability is shown under.

Statement of total costs

	Rs.
Direct Material cost	.....
Direct wages	.....
Direct Expenses	.....
"A" prime cost	.....
Add: Variable expenses	.....
"B" Marginal cost	.....
Add: Fixed overhead	.....
"C" Total cost	.....

Fixed and variable costs : Fixed costs are those costs which remain constant at all levels of production within a given period of time. In other words, a cost that does not change in total but become, progressively smaller per unit when the volume of production increases is known as Fixed Cost. It is also called "Period Costs".

E.g., Rent, salary, Insurance charges, etc.,

Variables costs are those costs which vary in accordance with the volume of output.

Absorption costing and Marginal Costing : Absorption costing is also termed as Traditional or Full Cost method. In this method, the cost of a product is determined after considering both fixed and variable costs. In absorption costing all costs are identified with the manufactured products.

Marginal costing is a technique where only the variable costs are considered while computing the cost of a product. The marginal cost of a product is in variable cost. In this method only variable costs are changed to the cost units. Fixed cost is written against contribution for that period.

Hence we can derive a formula for contribution as under:

Contribution = sale price - marginal cost

Standard costing : Standard costing is a specialised technique of costing. In this costing standard costs are pre-determined. Actual costs are compared with pre-determined costs. The variations between the two are noted and analysed. Measures are taken to control the factors leading to unfavourable variations. Standard costing serves as an effective tool in the hands of the management for planning, coordinating and controlling of various activities of the business.

Budgeting Vs Standard costing

	Budgeting	Standard costing
1	Budgeting considers operation of the business as a whole Hence it is more extensive.	Standard costing considers only the control of the expenses. Hence it is more intensive.
2	Budget is a projection of financial accounts.	Standard cost is the projection of cost accounts.
3	It does not involve with standardisation of products.	It requires standardisation of products.
4	Budgeting can be operated in part also.	It is not possible to operate this system in parts.
5	Budget can be operated without any standards.	Standard costing cannot exist without budget.
6	Budgets are maximum target of expenses above which actual expenses should not rise.	Standards are minimum targets which are to be attained by actual performance at specific efficiency level.

Cost Centres : Cost Centre is a location, person or item of equipment, in relation to which costs may be ascertained and used for the purposes of cost control.

Any raw material, labour or other input used by an organisation for the manufacturing process is cost which is allocatable as direct or indirect costs to cost centres.

1 **Direct Cost :** A cost which is allocated to a cost centre is a direct cost of that particular cost centre.

2 **Indirect Cost :** A cost which is apportioned to different cost centres on a suitable basis is an indirect cost of that particular cost centre.

Cost Category: Cost Categories are useful for organisations that require allocation of Revenue and Non-Revenue Items to parallel sets of Cost Centres. The examples of Cost Categories can be Region-wise, Grade-wise, Department-wise and so on.

For example, a company has three departments such as Marketing, Finance and Production. Each department has been identified as a cost centre. The wages paid to the respective workers of the department concerned is the direct cost of that particular cost centre(department). However, if the rent of the building in which the production departments are located is apportioned to the departments on a scientific basis, then it is termed as an indirect cost of the cost centre.

This is the example of use of Cost Categories. The Salesmen A, B and C can be Cost Centres under a Category Executive. Similarly, you can create a new Cost Category projects under which Cost Centres such as Airport construction, Road construction and Buildings may be created. So that the classification appears as following

Cost Categories	Departments	Executives	Projects
Cost Centres	Marketing	Salesman A	Airport Construction
	Manufacturing	Salesman B	Road Construction
	Finance	Salesman C	Buildings

Analysis using Cost Centres : Cost centre performance can be measured using volume or relative percentage. For example. Direct variable cost can be monitored as a percentage of sales or even as a percentage of cost of production if there are effective process controls. Indirect expenses can be monitored by creating a reasonable limit on the amount to be spent.

The following elements have to be kept in mind while measuring cost performance.

1 Overall objective of the business.

2 Changes in the business environment.

3 Ground realities.

Cost Centres in Tally : In Tally's cost centre allow for multi dimensional analysis of financial information. The cost centre feature in Tally allows you to allocate a transaction to a particular cost entre, gives the cost centre break-up of each transaction as well as details of transaction for each cost centre. A Profit and Loss Statement of every cost centre can also be obtained.

Cost Centre Reports : These are the following reports -

Category Summary : This report displays the summary of all the cost centers under a cost category.

Cost Centre Break-up: This report displays Ledger and Group summary information for the selected cost centre.

Ledger Break-up: This report displays the summary information of all Cost Centers for the selected Ledger.

Group Break-up: This report displays the summary information of all Cost Centers for the selected Group.

Inventory Accounting With Tally : Inventory accounting includes recording stock details, the purchase of stock, the sale of stock, stock movement between storage Location or Godowns, and providing information on stock availability. With Tally it is possible to integrate the inventory and accounting systems so that financial statements reflect the closing stock value from the inventory system.

The inventory system operates in much the same way as the accounting system.

- First you set up the inventory details, which is a similar operation to creating the chart of accounts although, in this case, there are No pre-defined set of stock groups.
- Second, you create the individual stock items, which is similar to setting up the ledgers.
- Finally, you are ready to use vouchers to record the various stock transaction.

The Inventory features comprises of configurations/ functionality pertaining to Inventory transactions and reports. The Inventory features are further sub-divided into seven sections:

General

Storage & Classification

Order Processing

Invoicing

Purchase Management

Sales Management

Other Features

Inventory Reports : Tally generates inventory reports based on vouchers entries made. You can use the customised reports to compare inventory data between different companies, periods of the financial year and so on. There are Inventory Books and Statements of Inventory.

Inventory Books : In inventory books there are stock item, group summary, stock transfers, physical stock register.

1 **Stock Items :** Stock item refers to goods in which you deal-that is, goods that you manufacture or trade(sell and purchase). It is the primary inventory entity. Similar to ledgers being used in accounting transactions - you have to use Stock Item in Inventory transactions. Therefore, Stock Items are important in Inventory like how Ledgers are important in Accounting.

2 **Units of Measure :** Stock items are mainly purchased and sold on the basis of quantity. The quantity is measured by units. In such a case, it is necessary to create the unit of measure. Units of measure can be simple units such as nos., meters, kilograms, pieces, or compound units, e.g. box of 10 pieces. Create the units of measure before creating the stock items.

3 **Stock Groups :** Similar to Groups in Accounting Masters - these are provided for the purpose of classification of stock items. Classification is done based on some common behavior. Grouping stock items enables easy identification and reporting of stock items in Statements. The group summary statement shows the closing balances of accounts under the selected group upto the current date. For example, items of a particular brand can be grouped together so that you can get the inventory details of all items of that brand. It is NOT necessary to group items.

4 **Stock Transfers :** Stock transfer report display entries made using stock journal vouchers.

5 **Physical Stock Register :** Physical stock register is used to record actual stock available, i. e. Stock found on conducting a stock check. It is not unusual to find a discrepancy between the actual stock and the computer recorded stock figure. If inventory vouchers have been configured to ignore physical stock differences, these physical stock vouchers will be useful for recording purposes.

Statements of Inventory : There are Godown, stock categories, stock query, reorder status, purchase bills pending, sales bills pending.

1 **Godown :** A place where stock items are stored is referred to as Godowns. You can specify where the stock items are kept, e.g. warehouse, shelf or rack, etc, and obtain stock reports for each Godown, and account for movement of stock between locations/ Godowns.

2 **Stock Categories :** This is a feature, which offers a parallel classification of stock items. Like Stock Groups, classification is done based on some similar behavior. The advantage of categorizing items that Tally allows you to classify stock items (based on functionality) together - across different stock groups, enabling you to obtain reports on alternatives or substitutes for a stock item.

3 **Stock Query :** Stock query allows you to obtain all information about a stock item. It displays all the necessary real time information about an item that you may require to help negotiate order with a customer on a single screen. It also display cost and price details of alternative items.

4 **Purchase Bills Pending :** Purchase bills pending displays all instance of incomplete purchases where goods may have been received, but not fully invoiced. It also displays instances of invoices received, but goods have not been received.

5 **Sales Bills Pending :** Sales bills pending displays all instance of incomplete sales where goods may have been delivered, but not fully invoiced. It also displays instances of invoices raised, but goods have not been delivered.

Budgeting systems, Scenario management and Variance analysis

Objectives: At the end of this lesson you shall be able to
- **understand the meaning of budget & budget manual**
- **know the budget period, classification of budgets and its types**
- **understand scenario management**
- **understand variance analysis.**

Introduction : Planning has become the primary function of management these days. Budgets are nothing but the expressions, largely in financial terms of management's plan for operating and financing the enterprise during specific periods of time.

A budget is a detailed plan of operations for some specific future period. It is an estimate prepared in advance of the period to which it applies. It acts as a business barometer as it is complete programme of activities of the business for the period covered.

Essentials of a Budget

a It is prepared in advance and is based on a future plan of actions

b It relates to a future period and is based on objectives to be attained

c It is a statement expressed in monetary and / or physical units prepared for the implementation of policy formulated by the management.

Budget manual : The budget manual is a written document or booklet which specifies the objectives of the budgeting organization and procedures.

The following are the important matters covered in a budget manual

1 A statement regarding the objectives of the organisation and how they can be achieved through budgetary control.

2 A statement regarding the functions and responsibilities of each executive regarding preparation of budget and execution of budget.

3 Procedures to be followed for obtaining the necessary approval of budgets.

4 Time table for all stages of budgeting.

5 Reports, statements, forms and other records to be maintained.

6 The accounts classification is to be employed.

Responsibility for Budgeting : There are two things responsible for budgeting. They are the budget controller and the budget committee.

1 **Budget controller :** The chief executive is ultimately responsible for the budget programme. But large part of the supervisory responsibility is deligated to an official designated as budget controller or budget director. The budget controller should have knowledge of the technical side of the business and should report direct to the president of the organisation.

2 **Budget committee :** The Budget controller will be assisted in his work by the Budget committee. The budget committee will consists of heads of the various departments as production, sales, finance, etc., Budget controller is the chairman of the committee. It will be the duty of the budget committee to submit, discuss and finally approve of the budget figures. Each Head of the Department will have his own sub-committee with executive working under him as in members.

Fixation of the Budget Period : Budget period means the period for which a budget is prepared and employed.

The budget period will depend upon the following:

1 The nature of the business and

2 The control techniques to be applied

For example, a seasonal industry will budget for each season. An industry requiring long periods to complete work will budget for 3 or 4 or 5 years. But Budget period should not be longer than that of what is necessary.

Budget Procedure

Determination of key factor : Key factor indicates whose influence must first be assessed in order to ensure the accomplishment of the functional budgets. Functional budget is related to different functions of a business, e.g., sales, production, purchases, cash, etc.,

The budget related to the key factor should be prepared first. Then the other budgets.

General list of key factors in different industries are given below:

	Industry	Key factor
1	Motor car	Sales demand
2	Aluminium	power

3	Petrolium Refinery	Supply of crudeoil
4	Electro optics	Skilled technicians
5	Hydral power generation	Monsoon

Making of forecasts : A forecast is the statement of facts likely to occur that may affect the flow of budget. Forecast is done before the budgeting starts. Forecasts are made regarding sales, production cost and financial requirements of the business.

Consideration of alternative combination of forecasts: Alternative combinations of forecasts are considered with a view to obtain the most efficient overall plan so as to achieve maximum profit.

Preparation of Budgets : After finalising the forecasts, the actual budgets will be prepared. One budget may be prepared on the basis of the other budget. For example production budget will be prepared on the basis of the sales budget.

Classification of Budgets

Budgets can be classified into three most common types

1 Time based Budget

2 Function based budget

3 Flexibility based budget

Time based budget : In terms of time, the budget can broadly be classified into four categories

a **Long term Budget :** A budget designed for a long period is termed as long term budget. The period may be from 5 to 10 years

b **Short term budget :** A budget designed for a period generally not exceeding 5 years is termed as short term budget

c **Current budgets :** A budget which is prepared for a very short period, say a month or a quarter is termed as current budget

d **Rolling Budgets :** A budget which is designed for a year in advance is termed as Rolling Budgets or Progressive Budgets

Function Based Budget

a **Sales Budget :** This budget forecasts total sales in terms of quantity, value, items, periods, area, etc.,

b **Production Budget :** This budget is based on sales budget. It forecasts quantity of production in terms of items, periods, areas, etc.,

c **Cost of Production budget :** This budget forecasts the cost of production, separate budgets are prepared for different elements of costs.

d **Purchase Budget :** This budget forecasts the quantity, and value of purchases required for production. It gives quantity -wise, money-wise and period - wise information about the materials to be purchased

e **Personnel Budget :** This budget anticipates the quantity of personnel required during a period for production activity

f **Research Budget :** This budget relates to the research works for improvement in quality of products or research for new products.

g **Capital Expenditure Budget :** This budget provides a guidance regarding the amount of capital that may be required for procurement of capital assets during the budget period

h **Cash Budget :** This budget forecasts the cash position by time period for a specific duration of time

i **Master Budget :** This is a summary budget incorporating all functional budgets in a capsule form.

Flexibility based Budget

a **Fixed Budget :** A budget prepared on the basis of a standard or a fixed level of activity is called a fixed budget. It does not change with the change in the level of activity

b **Flexible Budget :** A budget designed in a manner so as to give the budgeted cost of any level of activity is termed as a flexible budget.

Accounting Features : The Accounting Features consists of configurations / functionality, which generally affects accounting transactions and reports. The accounting features are further sub-divided into six sections, namely:

General

Out standings Management

Cost /Profit Centers Management

Invoicing

Budgets / Scenario Management

Other Features

Scenario Management: This is a management tool which displays of accounts and inventory related information, by selectively including certain types of vouchers. It is a forecasting tool which forecast the expenses using provisional Vouchers and includes them in the reports.

Scenario Analysis : Scenario planning is also called scenario analysis. It is a strategic planning method that some company use to make flexible long-term plans. Thus the scenario analysis is process of analyzing possible future events by considering alternative possible outcomes.

Several Scenarios are demonstrated in a scenario analysis to show possible future outcomes. It is useful to generate a combination of an optimistic, a pessimistic and a most likely scenario.

Variances : Variances can be computed for both costs and revenues.

The concept of variance is connected with planned and actual results. It is effect to the difference between those two on the performance of the company.

Variance Analysis : Variance analysis is a tool of budgetary control by evaluation of performance by means of variances between budgeted amount, planned amount or standard amount and the actual amount incurred/sold. Variance analysis is usually associated with explaining the difference (or variance) between actual costs and the standard costs allowed for the good output. For example, the difference in materials costs can be divided into a materials price variance and a materials usage variance. The difference between the actual direct labor costs and the standard direct labor costs can be divided into a rate variance and an efficiency variance. The difference in manufacturing overhead can be divided into spending, efficiency, and volume variances. Mix and yield variances can also be calculated.

Variance analysis helps the management to understand the present costs and thereafter control the future costs.

Concepts of Ratios, Analysis of financial statements

Objectives: At the end of this lesson you shall be able to
- **understand the concept of stock**
- **understand ratio analysis**
- **understand cash flow**
- **understand fund flow accounting**
- **explain the invoice.**

Introduction : In a trading business one of the current assets is stock. It represents goods owned by the company that are for sale to customers. The owners always aware of stock. Once a year the stock is counted and valued.

The system used for counting stock and evaluating the value of the stock is known as Inventory system. When business has a more extensive stock, Inventory system is used. Stock is controled with the Inventory system.

Closing stock and its valuation : Closing stock means closing stock of raw materials or goods manufactured. The closing stock must be valued and an entry passed at the end of the year.

– Suppose an article is purchased for Rs.100. If the article remains unsold at the end of the year, it will be included in the closing stock at Rs.100 even if the selling price is more

– But if the article can now be sold at Rs.95only, it should be included in the closing stock at Rs.95 only

– Goods which cannot be sold at all should not be included in the closing stock.

Casting stock Value : Stock value can be calculated in four methods. They are

1 First in First out (FIFO)

2 Last In First Out (LIFO)

3 Average

4 Base stock method

FIFO : In this case the earliest lots are exhausted first. The stock on hand is out of the latest consignments received and is valued accordingly.

Suppose following lots were received:

16th October 200 units @ Rs.10

20th November 300 units @ Rs.11

15th December 250 units @ Rs.11.50

The closing stock consists of 300 units . The value will be

250 units @ Rs.11.50	Rs.2,875
50 units @ 11.00	Rs. 550
Total	Rs.3,425

LIFO : In this case the latest consignments are used first. Hence the closing stock is supposed to be out of the earliest lots on hand. For the above example, the stock will be valued at Rs.3,100, as under:

200 units @ Rs.10	Rs.2,000
100 units @ Rs.11	Rs.1,100
	Rs.3,100

Average Method : In this case all the lots are merged together and value of the closing stock is calculated accordingly. The average may be simple or weighted.

Simple average Method : In this method, the average price for single unit is

$$= \frac{Rs.10 + Rs.11 + Rs.11.50}{3}$$

$$= \frac{32.50}{3}$$

$$= Rs.10.83$$

The value for closing stock is = 300 x 10.83 = Rs.3,249

Weighted Average method : Weighted average method is most suitable, since the quantities are also taken into account.

In this method the average price for single unit is Rs.10.90, calculated as under.

Number of units	**price**	**Amount**
	Rs.	**Rs.**
200	10	2,000
300	11	3,300
250	11.50	2,875
750	Total	8,175

Unit cost = $\frac{8,175}{750}$

= 10.90

The value of 300 units = 300 x 10.90

= Rs. 3,270

Base Stock Method : In this method, the minimum stock carried by the factory is valued at the price originally paid for it. The excess of actual stock over the minimum level is valued according to the current cost, calculated in one of the three methods given above.

Suppose the minimum stock is 200 units and the original price paid was Rs.8 per unit.

The value of stock of 200 units @ Rs.8 = Rs.1,600

The value of remaining 100 units is calculated with the unit price of Rs.11.50 or Rs.11.00 or 10.83 or 10.90

Statement of stock : M/s. Nanda & Bose close their financial books on 30th June 2001. Stock taking continues for two weeks after this date. In 2001, the value of stock came to Rs.20,500, without making adjustments for the following:-

1 Purchases made during the two weeks after 30th June 2001 were Rs.500

2 Sales made during these two weeks amounted to Rs.3000. The firm makes a gross profit of 33 1/3% on sales. Find out the value of closing stock on 30th June, 2001.

For this case the statement of stock can be calculated as under:

Statement of stock

Stock on 30th June 2001

	Rs.
Sales	3,000
Less 33 1/3% on sales	1,000
Sales at cost	2,000
Value of stock two weeks after 30th June 2001	22,500
Less:purchases during 2 weeks	500
Value of stock on 30th June 2001	22,000

Inventory control and Reordering : Inventory control system always monitors the availability of stock. In any business minimum stock should be maintained for uninterupted sale. If the minimum stock level falls, then the purchase order for that product is to be made.

The minimum stock level below which purchase order is to be made is known as reorderlevel. If the stock level falls below the reorder level purchase order is proposed.

Example

Item No.	Description	Stock	Reorder level	Reorder yes/no
10283	REXONA BIG	125	50	N
10284	REXONA SMALL	25	50	Y
10285	CINTHOL NEW	85	75	N
10286	CINTHOL OLD	15	25	Y
10287	VICCO PASTE 100	10	15	Y
10288	VICCO PASTE 50	00	15	Y
10289	VICCOTURMERIC	15	27	N

Purchase order should be made for the items 10284, 10286, 10287, 10288. Hence the purchase order can be proposed using the inventory control system.

Ratio Analysis : Ratio Analysis is used to evaluate various aspects of a company's operating and financial performance such as its efficiency, liquidity, profitability and solvency.

In Accounting, there are many standard ratios used to try to evaluate the overall financial condition of a corporation or other organization. It are also compared across different companies in the same sector to see how they stack up, and to get an idea of comparative valuations.

Types of Ratios :

1 **Financial Ratios :** These are categorized according to the financial aspect of the business which the ratio

measures. It allow for comparisons between companies, between industries, between different time periods for one company, between a single company and its industry average.

2 **Liquidity Ratios :** It is measure that the availability of cash to pay debt.

3 **Activity Ratios :** It is measure how to quickly a firm converts non-cash assets to cash assets.

4 **Debt Ratios :** It is measure the firm's ability to repay long-term debt.

5 **Profitability Ratios :** It is measure the firm's use of its assets and control of its expenses to generate an acceptable rate of return.

6 **Market Ratios :** It is measure the investor response to owning a company's stock and also the cost of issuing stock.

Cash Flow Accounting : In accounting cash flow is the difference between the amount of cash available at the opening balance (beginning of a period) and the amount at the closing balance (end of that period).

When Cash is coming in from customers or clients who are buying your products is called accounts receivable and when the cash is going out of your business in the form of payments for expenses like rent, loan payment is called accounts payable.

The net cash flow of a company over a period is equal to the change in cash balance over this period : positive if the cash balance increases, negative if the cash balance decreases. The total net cash flow is the sum of cash flows that are classified in three areas:

1 Operational Cash Flows : Cash received or expended as a result of the company's internal business activities. It includes cash earnings plus changes to working capital.

2 Investment Cash Flows : Cash received from the sale of long-life assets, or spent on capital expenditure.

3 Financing Cash Flows : Cash received from the issue of debt and equity, or paid out as dividends, share repurchases or debt repayments.

In financial accounting, a cash flow statement is also known as statement of cash flows which is a financial statement that shows how changes in balance sheet accounts and income affect cash and cash equivalents, and breaks the analysis down to operating, investing and financing activities. The cash flow statement is concerned with the flow of cash in and out of the business.

Fund Flow Accounting : These statements give the information of funds on a particular date. The purpose of preparation of funds flow statements is to know about from where funds are coming and where being invested. The fund flow stalemates is generally prepared from the data identifiable and profit and loss account and balance sheets. Fund Flow statement is also called as sources and application of funds. It shows the detail of funds business received from sources and the amount of funds the business used for different purpose in the year.

Invoice : While making a sale, the seller prepares a statement giving the particulars such as the quantity, price per unit, the total amount payable, any deductions made and shows the net amount payable by the buyer. Such a statement is called invoice.

An invoice is a statement of list of goods with their quantity and price.

An invoice is a business document which is prepared by sellers and given to buyers. Usually the invoice are prepared in triplicate one copy will be issued to the buyer. The after two copies will be retained by the seller.

Inward Invoice : To the buyer, he calls the invoice as 'inward invoice'. He enters the details of the invoice in his purchase book.

Outward Invoice : To the seller, he calls the invoice as 'outward invoice'. He enters the details of the invoice in his sales book.

Details of the Invoice : The invoice should give the following details:

1 Name and address of the seller

2 Name and address of the buyer

3 Invoice Number

4 The Date

5 Quantity, description, unit price, amount of the goods sold

6 Trade discount and cash discount, if necessary

7 Expenditure

8 Net amount

9 Signature of the invoicing authority

10 E. & O. E. at the left bottom corner of the invoice

E. & O. E. means Errors and Omission Excepted. This indicates that if errors and omissions are found out, the matter can be reported and settled and the seller under takes to correct them.

Tax processing in Tally

Objectives: At the end of this lesson you shall be able to
- **explain statutory & taxation**
- **explain direct & indirect taxes**
- **explain VAT & service tax processing**
- **explain TDS, TCS, FBT taxes**
- **explain GST.**

Statutory & Taxation : The Statutory & Taxation features comprises of configurations/functionality pertaining to statutory compliances available in Tally.ERP9. The Statutory features are country specific and strictly depends upon the Country selected in the Company Creation screen. The following features are available, when India is selected in the Statutory Compliance for field in the Company Creation screen.

- Excise
- Value Added Tax
- Service Tax
- Tax Deducted at Source
- Tax Collected at Source
- Payroll
- Income Tax
- GST

Tax : The Tax is a financial charge upon an individual (called taxpayer) by a country / state or the functional equivalent of a state such that failure to pay, is punishable by law.

Taxes are two types : direct or indirect taxes.

Direct Taxes : Direct Taxes are taxes collected by government, directly from tax payers, through levies such as income tax, wealth tax and interest tax. In the case of direct taxes, the incidence and burden of paying the tax falls on the same person.

TDS : Tax Deducted at Source (TDS) is one of the modes of collecting income tax in india. The buyer (deductor) deducts the tax from the payment made to the seller (deductee) and remits the tax to the Income Tax Department within the stipulated time.

The buyers (Corporate and Non-Corporate) make payments (such as Salary, Rent, Dividends, Professional Fees, Commission, etc) to the sellers (Services) and deduct the requisite amount from such payments towards tax.

These taxes are on Immovable Property, Contractor, Rent of Machinery, Rent of Land and Building.

The buyer files the TDS returns containing details of the seller and the bank, where the Tax Deducted at Source amount is deposited to the Income Tax Department.

Tax Collected at Source : Tax Collected at Source (TCS) is income tax collected by seller in India from payer on sale of certain items. The seller has to collect tax at specified rates from the payer who has purchased these items :

- Alcoholic liquor for human consumption
- Tendu leaves
- Timber obtained under a forest lease
- Timber obtained by any mode other than under a forest lease
- Any other forest produce not being timber or tendu leaves
- Scrap
- Parking lot
- Toll plaza
- Mining and quarrying

The TCS on the above mentioned items vary from 1% to 5%.

FBT : Fringe Benefits Tax (FBT) is taxation of most which are generally non-cash employee benefits. A new tax was imposed on employers by India's Finance Act 2005 was introduced for the financial year commencing April 1, 2005. The fringe benefit tax was temporarily suspended in the 2009 Union budget of India by Finance Minister Pranab Mukherjee.

The following items were covered:

- Employer's expenses on entertainment, travel, employee welfare and accommodation. The definition of fringe benefits that have become taxable has been significantly extended. The law provides an exact list of taxable items.
- Employer's provision of employee transportation to work or a cash allowances for this purpose.
- Employer's contributions to an approved retirement plan (called a superannuation fund).

- Employee stock option plans (ESOPs) have also been brought under fringe benefits tax from the fiscal year 2007-08.

Indirect Taxes : Indirect Tax, imposed on commodities, is indirectly borne by the people. It includes value added tax, sales tax, customs duty, excise duty. In the case of indirect taxes, the person on whom the incidence to pay the tax falls is different from the person who carries the burden of paying the tax.

Service Tax

Service Tax is a tax imposed by Government of India on services provided in India. The service provider collects the tax and pays the same to the government. It is charged on all services except the services covered in the negative list (Section 66d of Finance Act'1994) of services & services covered under Mega Exemption Notification (Notification NO. 25/2012 ST dated 20.06.2012). The current rate is 12.36% on gross value of the service.

Dr.Raja chelliah committee on tax reforms recommend the introduction of service tax. Service tax had been first levied at a rate of five per cent flat from 15 July 1994 till 13 May 2003, at the rate of eight percent flat w.e.f 1 plus an education cess of 2% thereon w.e.f 10 September 2004 le services provided by service providers. The rate of service tax was enhanced to 12% by Finance Act, 2006 w.e.f 18.4.2006. Finance Act, 2007 has imposed a new secondary and higher education cess of one percent on the service tax w.e.f 11.5.2007, increasing the total education cess to three percent and a total levy of 12.36 percent. The revenue from the service tax to the Government of India have shown a steady rise since its inception in 1994. The tax collections have grown substantially since 1994-95 i.e. from Rs. 410 crores in 1994-95 to Rs.132518 crores in 2012-13. The total number of Taxable services also increased from 3 in 1994 to 119 in 2012. However, from 1 July 2012 the concept of taxation on services was changed from a 'Selected service approach' to a 'Negative List regime'. This changed the taxation system of services from tax on some Selected services to tax being levied on the every service other than services mentioned in Negative list.

Service Tax Return

According to Rule 5 of Service Tax Rules, 1994, records include computerized data and means the record as maintained by an assessee in accordance with the various laws in force from time to time. Records maintained as such shall be acceptable to Central Excise Officer. Every assessee is required to furnish to the Central Excise Officer at the time of filing his return for the first time a list of all accounts maintained by the assessee in relation to Service Tax including memoranda received from his branch offices. This intimation may be sent along with a covering letter while filing the service tax return for the first time.

Invoice

Rule 4A prescribes that taxable services shall be provided and input credit shall be distributed only on the basis of a bill, invoice or challan. Such bill, invoice or challan will also include documents used by service providers of banking services (such as pay-in-slip, debit credit advice etc.) and consignment note issued by goods transport agencies. Rule 4B provides for issuance of a consignment note to a customer by the service provider in respect of goods transport booking services.

Value Added Tax : The Government of India has, after committing to the World Trade Organization regime, decided to modernize and streamline its indirect taxation, in the light of the experience of other WTO member countries. The Government has availed of the services of the international management consulting firm for drafting of rules, procedures and forms for introduction of VAT. VAT is prevalent in over 140 countries including India. Introduction of VAT would be a historic reform of the domestic trade tax system. It is expected to facilitate the states and union territories to transit successfully from the erstwhile sales tax system to modern domestic system.

A Value Added Tax (VAT) is applies the equivalent of a sales tax to every operation that creates value. The example is a Toy manufacturer company imported plastic. That company will pay the VAT on the purchase price, remitting that amount to the government. The company will then use the plastic into a toy, selling the toy for a higher price to a wholesale distributor. The company will collect the VAT on the higher price, but will remit to the government only the excess related to the "value added". The wholesale distributor will then continue the process, charging the retail distributor the VAT on the entire price to the retailer, but remitting only the amount related to the distribution mark-up to the government. The last VAT amount is paid by the retail customer who cannot recover any of the previously paid VAT. For a VAT and Sales Tax of identical rates, the total tax paid is the same, but it is paid at differing points in the process.

VAT is usually administrated by requiring the company to complete a VAT return, giving details of VAT it has been charged (input tax) and VAT it has charged to others (output tax). The difference between output tax and input tax is payable to the Local Tax Authority.

If input tax is greater than output tax the company can claim back money from the Local Tax Authority.

Mechanism of tax credit:

For the VAT auditors, the knowledge of tax credit is utmost important. Therefore it is necessary to understand the mechanism of tax credit.

WHAT is Tax Credit ?

1 It is a salient feature of VAT.

2 It is available to registered Purchasing Dealer.

3 Availability of the credit of tax paid on purchases of taxable goods.

4 Tax credit is available at the point of purchases.

5 Purchases of goods should be intended for the specific purposes.

6 Original Tax Invoice is must stating separate amount of tax charged.

7 Tax credit is not dependent or related either sale of very goods or it is used in manufacturing.

8 Tax Credit is adjustable against payment of tax / liability of tax.

GST (Goods and Services Tax)

- Goods and Services Tax (GST) is an indirect tax levied in India on the supply of goods and services.
- GST has been introduced to replace multiple indirect taxes levied by State and Central Governments in order to simplify the indirect tax.
- GST is levied at every step in the production process, but is refunded to all parties in the chain of production other than the final consumer.
- GST is a comprehensive Value Added Tax (VAT) on goods and services.
- France was the first country to introduce this system in 1954. Today it has spread to over 140 countries.
- Comprehensive dual GST has been implemented in India Since 1st July 2017.

GST means different things to different stakeholders. Businesses registered as regular dealers need to file their GSTR-1 on a monthly basis if their aggregate turnover exceeds 1.5 Cr. Businesses with aggregate turnover less than 1.5 Cr have to do GSTR-1 return filing on a quarterly basis. Also, both the businesses need to file their GSTR-3B on a monthly basis.

On the other hand, composite dealers have to file GSTR-4 on a quarterly basis. Also, going forward, as e-Way bill becomes mandatory for interstate and intrastate movement of goods worth Rs. 50,000/-

Components of GST?

There are 3 taxes applicable under GST: CGST, SGST & IGST.

CGST: Collected by the Central Government on an intra-state sale (Eg: Within the same State)

SGST: Collected by the State Government on an intra-state sale (Eg: Within the same State)

IGST: Collected by the Central Government for inter-state sale (Eg: one state to another state)

Tally.ERP 9 ensures you generate GST invoices and transactions as per the GST format.

Able to file GSTR-1, GSTR-3B and GSTR-4 on your own by exporting data to the Excel Offline Utility tool or in JSON format as per the GST portal. The unique error detection and correction capability ensures that you file returns accurately.

When it comes to e-Way Bills, Tally.ERP 9 helps you to easily generate and manage e-Way Bills. You can capture all the required information at invoice level itself, export the data in JSON format and upload the data in the e-Way portal to generate the e-Way Bill.

In the year 2005, VAT was introduced to overcome cascading affect (tax on tax). While VAT did eliminate the cascading tax effect on the indirect taxes within a state, the cascading effect of other indirect taxes across the country, still remained. For example, the Central Sales Tax (CST) applicable on interstate trade was non-creditable, leading to a break in the input credit chain. Similarly, a manufacturer charging Excise Duty on sale to a dealer caused the chain to break. This uncreditable tax found its way into the product cost.

GST on the other hand, allows for seamless flow of tax credit, and eliminates the cascading effect of all indirect taxes across the supply chain from manufacturers to retailers, and across state borders.

A quick comparison of the taxes one paid in the previous regime and in the current regime, will able to understand the aspect of GST vs VAT clearly.

- **Previous Regime**-Taxes paid by the dealer (Excise) to the manufacturer is added to the cost. When the dealer sells down the chain, VAT keeps getting charged on the sum of actual product cost + excise component, and the VAT keeps getting levied at every point of sale, till it reaches the end customer.
- **GST Regime**- Taxes paid by dealer (CGST + SGST) to manufacturer is not added to cost. This is because GST allows the dealer to set off the tax liability of CGST + SGST. This is one of the fundamental features of GST, which allows seamless credit from manufacturer to dealer, and eliminates the cascading effect of taxes.

At the 26th GST Council meeting, it has been decided to implement he inter-state e-way bill from 1st April, 2018. For intra-state movement, the e-way bill will be rolled out in a phased manner starting from 15th April, 2018, such that all states are covered by 1st June, 2018.

Eway bill - Introduction

In its 22nd meeting, the GST Council decided and recommended that the e-way bill under GST shall be introduced in a staggered manner from 1st January, 2018, and will be rolled out nationwide from 1st April, 2018.

However, at the recently concluded 24th GST Council meeting held on 16th December, 2017, it was announced that the e-way bill will be launched from the 1st of February, 2018 – a full two months ahead of the earlier plan.

The GST Council, reviewed the readiness of the hardware and software required for the nationwide rollout of e-way bill, and has announced the renewed date, post discussions with all the States.

The E-Way Bill is applicable for any consignment value exceeding INR 50,000. Even in case of inward supply of goods from unregistered person, E-Way Bill is applicable. The E-Way Bill needs to be generated before the commencement of movement of goods. Form GST EWB-01 is an E-Way Bill form. It contains Part A, where the details of the goods are furnished, and Part B contains vehicle number. For multiple Consignments the transporter should generate a consolidated E-Way Bill in the Form GST EWB 02 and separately indicate the serial number of E-Way Bill for each of the consignment.

Upon generation of the E-Way Bill, on the common portal, a unique E-Way Bill number called 'EBN' will be made available to the supplier, the recipient and the transporter.

If the recipient of goods doesn't communicate acceptance or rejection within 72 hours, it will be deemed as accepted by the recipient. The facility of generation and cancellation of E-Way Bill will be made available through SMS.

HSN stands for Harmonized system of Nomenclature which was developed by world customs organisations (WCO) with the vision of classifying goods all over the world in a systematic manner.

HSN Contains six digit uniform code that classifies 5000 + products and which is accepted world wide. HSN code describes the commodity/product.

India has already been using HSN system in the central excise and customs regime.

Tax payers whose turnover is below Rs.1.5 crores are not required to mention HSN code in their invoices. The list of HSN codes are available in public domain.

The E-Way Bill format in GST comprises of 2 parts – Part A and Part B.

GST E-Way Bill Format

FORM GST EWB-01

(See Rule 138)

E-Way Bill

PART A

A.1 : GSTIN of Recipient **:** Mention the GSTIN number of the recipient.

A.2 : Place of Delivery : Mention the Pin Code of the place where goods are delivered.

A.3 : Invoice or Challan Number : Mention the Invoice or Challan number which the goods are supplied.

A.4 : Invoice or Challan Date : Mention the Invoice or Challan Date which the goods are supplied.

A.5 : Value of Goods: Mention the consignment value of goods.

A.6 : HSN Code :enter the HSN code of goods which are transported. If your turnover is up to INR 5 crores, you need to mention the first 2 digits of HSN code. If it is more than INR 5 crores, 4 digits of HSN code are required.

A.7 : Reason for Transportation : The reason for transportation is pre-defined and you need to select the most appropriate option from the list.

Code Description

1 Supply

2 Export or Import

3 Job Work

4 SKD or CKD

5 Recipient not known

6 Line Sales

7 Sales Return

8 Exhibition or fairs

9 For own use

10 Others

A.8 : Transport Document Number : This indicates either one of the Goods Receipt Number, Railway Receipt Number, Airway Bill Number or Bill of Lading Number.

PART-B

B.1 : Vehicle Number : the vehicle number in which goods are transported needs to be mentioned. This will be filed by the transporter in the common portal.

FORM GST EWB-02

(See Rule 138)

Consolidated EWay Bill	
Number of EWay Bills:	:
EWay Bill Number	

Utilities

Objectives: At the end of this lesson you shall be able to
- **utilities in Tally**
- **split company data**
- **export master data**
- **import master, vouchers**
- **enable Tally vault password.**

Utilities & Others @ Tally.ERP9

- **Back Up / Restore**
- **Split Company Data**
- **Tally.ERP 9 Vault**
- **Exporting Master Data**
- **Importing Data**
- **Consolidation of Accounts**
- **Password or Security Control**
- **Enabling Cheque Printing**
- **Credit Limits**
- **Bills of Materials (BOM)**
- **Re-Order Levels and Re-Order Quantity**
- **Price Levels or Price List**
- **Interest Calculations**
- **Voucher Classes & Voucher Types (Creations)**
- **Point of Sale (POS Invoicing)**

Splitting Company Data based on Financial Years

Prerequisites for splitting company data

Before splitting the data, the user must ensure that:

- A backup of the data exists.
- All unadjusted forex gains/losses have been fully adjusted by recording journal entries.
- No purchase/sales bills are due. Check the **Profit & Loss A/c** and inventory statements (purchase/sales bills pending). You have to account them in the respective party accounts or in the bills pending account.
- The company data is verified to ensure that no errors occur during splitting using the **Verify Company Data** option.

To split the company data

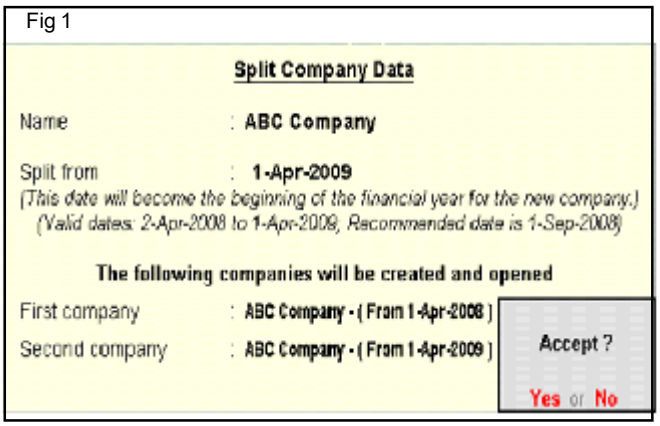
1 Go to **Gateway of Tally** > **F3 : Cmp Info** . > **Split Company Data** > **Select Company** .

2 Select the required company from the **List of Companies** .

3 Enter the required date in the **Split from** field.

The **Split Company Data** screen appears as shown below: (Fig 1)

Fig 1



4 Press **Enter** to split the company data.

Important points to remember:

- The **Split from** date is based on the existing data, and is considered as the beginning of the current financial year.
- Once the company data is split, two separate companies will be created and opened, without any changes to the original data.
- After the split, all the companies act as separate companies. You can make entries, display reports and, alter any data in these companies.

Export Data from Masters in Tally.ERP 9

To export masters

1 Go to **Gateway of Tally > Display > List of Accounts.**

2 Click **E : Export** to open the **Export Report** .

3 Press **Backspace** to configure the export options.

4 Select the **Language** .

5 Select the **Format** .

6 Enter the **Export Location** .

7 Enter the **Output File Name** .

8 Select the **Type of Masters** to be exported.

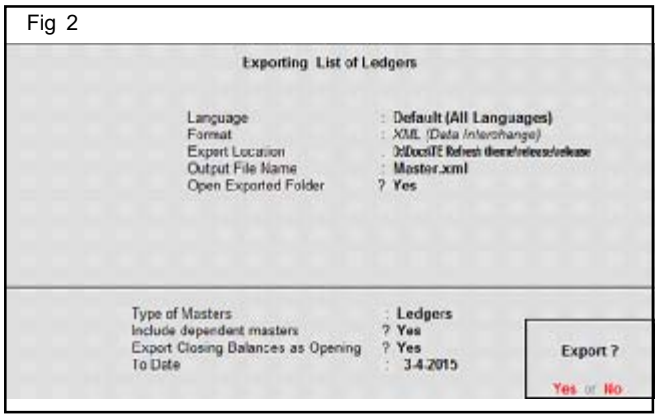
9 Set the option **Include dependent masters?** to **Yes** .

10 Enable **Export Closing Balances as Opening** , if required. The **Export Report** screen appears as shown below: (Fig 2)

11 Press **Enter** to export.

The exported file is saved in the location specified.

Fig 2



Importing Masters

Import masters previously exported from Tally.ERP 9 in XML format into the data of a company. Before importing, ensure that all the same features are enabled and disabled in the **Company Features** as the company from which the data is being imported.

For example, if you want to import masters created in ABC Company to a new company - XYZ Company. The options **Maintain stock categories** and **Maintain batch-wise details** must be enabled in ABC company before exported the masters. To import masters into XYZ Company, you need to ensure that both the options **Maintain stock categories** and **Maintain batch-wise details** are enabled in XYZ Company before importing.

To import masters

1 Go to **Gateway of Tally > Import Data > Masters** .

2 Enter the name of the .xml file to be imported, if the file is located in the Tally.ERP 9 application folder.

> **Note: By default, the export location is the Tally.ERP 9 installation folder, which is also the default import location. Therefore, it is not required to specify the file path during import. If the path is other than the installation folder, you must specify the exact location path, for example, C:\Export_Files\XML\Master.xml or C:\Export_Files\XML\DayBook.xml**

3 Select the **Behaviour** to define the method by which the existing entries in the company will be treated.

The **Import Masters** screen appears as shown below: (Fig 3)

Fig 3

4 Press **Enter** to import.

Importing Vouchers

Importing vouchers from one company to another in Tally.ERP 9 could be due to the following reasons:

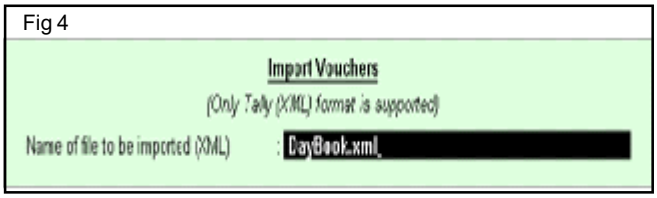
- Data corruption/loss.
- Migrating into a later release.
- Importing data from third party.

To import vouchers

1 Go to **Gateway of Tally** > **Import Data** > **Vouchers** .

2 Enter the name of the .xml file to be imported, in the **Import Vouchers** screen, as shown below: (Fig 4)

Fig 4



Note : By default, the export location is the Tally.ERP 9 installation folder, which is also the default import location. Therefore, it is not required to specify the file path during import. If the path is other than the installation folder, you must specify the exact location path, for example, C:\Export_Files\XML\Master.xml or C:\Export_Files\XML\DayBook.xml

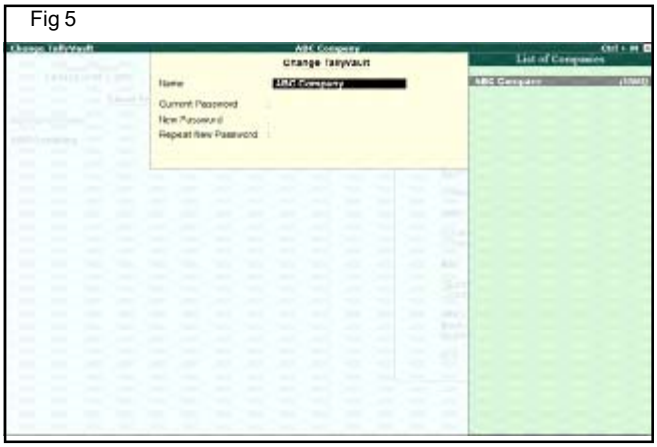
3 Press **Enter** to import.

Enabling Tally Vault

The user can enter the Tally Vault password while creating the company or execute the following steps to provide the TallyVault password for existing companies.

1 Go to **Gateway of Tally press F3 : Company Info > Change TallyVault** (Fig 5)

Fig 5



2 In the **Change Tally Vault** screen select the required company from the List of Companies.

3 Enter the Password in the New Password field. Tally.ERP 9 displays the strength of the password entered depending on the combination - Alphabets, Numbers and Special Characters.

4 Re-enter the password to confirm in the Repeat New Password field.

5 Accept to Change the Tally Vault password.

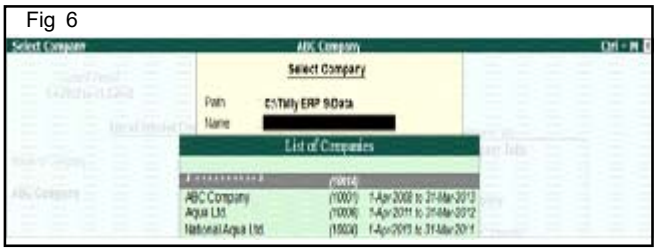
6 Tally.ERP 9 displays a message Created New Company followed by the new Company Number, press any key to return to Company Info menu.

Once the company data is encrypted the Name of the Company and Financial Year will not be visible in the Select Company screen.

7 In the Company Info, press Select

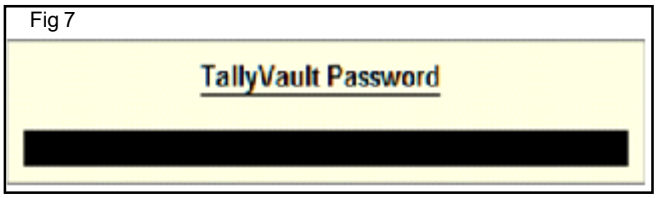
8 The Select Company screen with the encrypted company is displayed as shown in Fig 6.

Fig 6



9 Select the encrypted company, Tally.ERP 9 will prompt the user to provide the TallyVault password. (Fig 7)

Fig 7



10 Provide the required password and the company data is available for use in a readable format.

To enable Tally Vault while creating a new company, you must provide the TallyVault password and repeat the password, the new company created will be secured using Tally Vault.

Creating Users, Backup and restore

Objectives: At the end of this lesson you shall be able to
- **create users and passwords**
- **backup and restore data.**

You can create users, assign security levels, restrict/allow remote access and local TDLs for the users created.

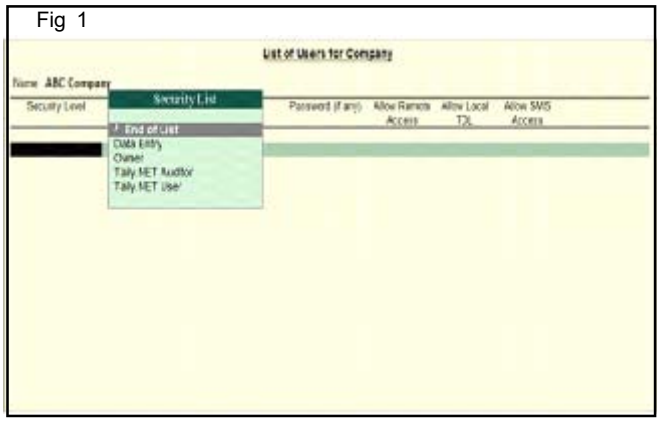
To create the user and assign a password execute the following steps:

Go to **Gateway of Tally > F3: Company Info > Security Control**

1 Select Users and Passwords

2 The List of Users for Company screen is displayed as shown Fig 1.

Fig 1

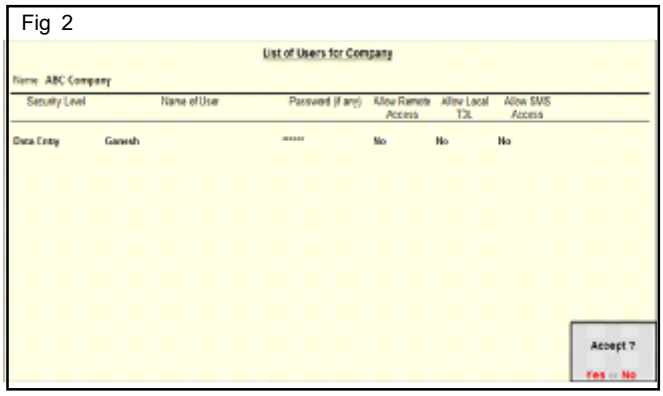


3 Select the required Security Level from the Security List

4 Enter the user's name in the Name of User field.

5 Enter the password in Password (if any) field. (Fig 2)

Fig 2



Backup & Restore data In Tally.erp 9

Tally.erp 9 provides the mechanism of taking a back up data from on store medium into another storage medium.You can take a backup on the local drive or in an external media.

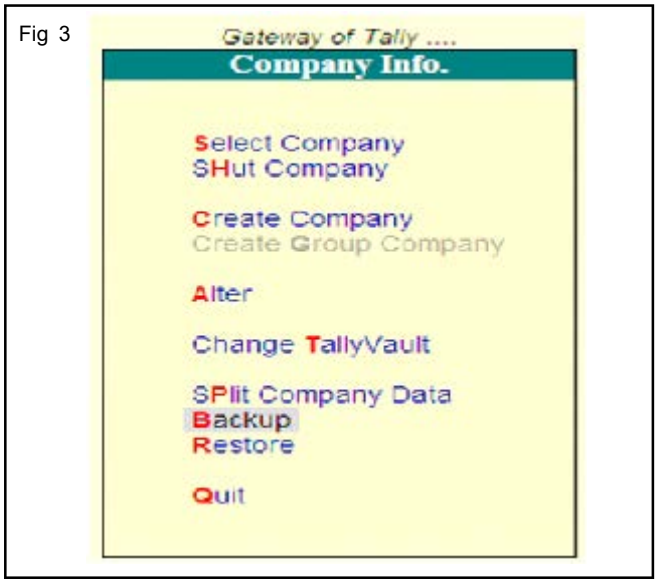
Taking backup is easy in tally and you just provide the source and destination location of the backup data.

You can create a main backup directory with subdirectories to take daily backup, for example, you can create a directory named tally backup with the subdirectories named.

Perform the following steps to take the backup of data in Tally ERP 9

- Click the F3 Comp info: button on the button bar in the gateway of tally screen the company info menu appears
- select the backup option from the company info menu as shown in Fig 3

Fig 3



- The backup companies on disk screen appears containing two main option source and destination
- Type the path of the company beside the source option to specify the location from where you want to take.

To do Auto Backup In Tally ERP 9

Data on the computer is dangerous from different types of threat and any data loss cannot be recovered hence there is need to store date at a different location by taking a backup.

To activate a backup option of auto backup

1 Go to a gateway of Tally > Alt + F3: comp info > Alter and select company from the list.

2 Company Alternation screen appears.

3 Enable auto backup? set to Yes

4 Accept the company

5 Press ESC to come on Gateway of Tally Screen

6 GO to Gateway of tally F12: Configuration > Date Configuration

7 In location of auto backup files type the path in which path you want to get back up D:\autobackup

8 Accept the data configuration Screen.

9 Message appears that do you want to restart Tally for the change to have effect press Y.

10 Now when you close the Company Tally take the backup automatically.

To restore Auto Backup Data

1 Go to Gateway of Tally > ALt + F3: Comp info >Restore.

2 In destination, field type the path in which you want to restore the back up data.

3 In source, field type the path: D:\autobackup.

4 In auto Backup section list of auto, backup appears select company from the list.

5 In backup version select the latest backup.

Multilingual capability in Tally

Objectives: At the end of this lesson you shall be able to
- **list of accounts/ledgers**
- **configuration in tally**
- **multilingual capability in tally.**

List of Accounts : Tally.ERP9 gives great flexibility in list of accounts which displays the list as groups in alphabetical order. The groups are in bold and begin on the extreme left. The sub-groups are also in bold and the ledger accounts are in italic and in the lowest level. The report is drill down and if press the enter key and then it display ledger Alteration (Secondary) screen through the ledger accounts.

Configuration in Tally : Tally.ERP9 allows you to modify these when your requirements to change the configurations.

In Configuration consists of the following menus.

- **General :** with this option you can configure the Country Details, Style of Names, Style of Dates.
- **Numeric Symbols :** with this option you can configure the number styles.
- **Accts / Inventory Info. :** with this option you can configure the details in Accounts Masters and Inventory Masters.
- **Voucher Entry :** with this option you can configure the vouchers entries in Accounting and Inventory Vouchers.
- **Invoice / Orders Entry :** with this option you can configure the invoice, delivery notes, sales & purchase orders.
- **Payroll Configuration :** with this option you can configure the statutory details, passport details, contract details and deactivated employees.
- **Banking Configuration :** with this option you can configure the settings related to Bank reconciliation statement.
- **Printing :** with this option you can configure the printing parameters of a voucher, invoice and statement layouts before final printing.
- **E-Mailing :** with this option you can configure the e-mailing facility.
- **Data Configuration :** with this option you can configure the path where the language, data and configuration files reside.
- **Advanced Configuration :** with this option you can configure the Client / Server, ODBC, Connection, Log, Tally.NET.
- **Licensing :** with this option you can configure the update, surrender, reset license.

Multilingual Capability in tally : Tally.ERP9 allows you to record, view, print information in any one of the 9 Indian language (like Hindi, Gujarati, Punjabi, Tamil, Telugu, Marathi, Kannada, Malayalam and Bengali), besides few international languages such as Arabic, Bahasa Indonesia, Bahasa Malayu etc. Tally enables you to enter data in one language and have it transliterated into different languages.

Some others features :

- It is a user friendly.
- It offers concurrent multilingual support.
- It maintain your books of accounts while the data is accepted, sorted, maintained, displayed and printed in any one of the language.
- It generate bill, invoices, vouchers, ledgers, receipts, reports, purchase orders or delivery notes in the language of your choice after entering data for in any one of the specified languages.
- It has easy to use keyboards layouts - inscript and phonetic.

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**
- **explain capabilities required and technology issue 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 Asssociate 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 it 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 filpkart, 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 descisions.

  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

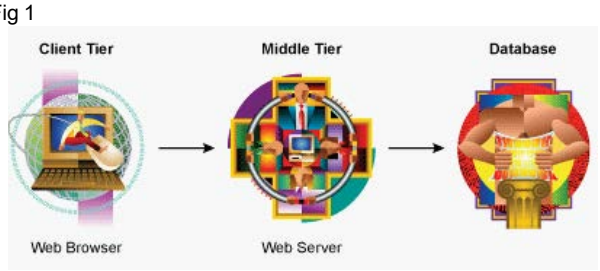
- Operational benefits of e commerce include reducing both the time and personnel required to complete business processes, and reducing strain on other resources. It's because of all these advantages that one can harness the power of ecommerce and convert a business to E Business by using powerful E Commerce solutions made available by E Business solution providers.

Technology Requirements for E Commerce

The requirements for E Commerce needs Web based application with an HTML front end compatible with a variety of browsers. The application will require a database to store user transactions (such as items ordered by the user) and to list items available through the electronic store.

The application must also include a middle tier-the Web server and scripts executed by the server-to process requests sent from Web browsers. A Web browser will send HTTP requests to the middle tier. The middle tier will then retrieve information stored in the database, process it appropriately, and send a reply back to the client. (Fig 1)

Fig 1



Database

An electronic database is utilized to store data associated with E Commerce applications. Such information might include information pertaining to registered users, goods and services, and records of transactions effected by visitors to the site.

We have two alternatives for storing this information:

- Employing flat files to store the data in text files.
- Utilizing a relational database system such as MySQL, Oracle or SQL Server.

Flat files are strongly discouraged because they require excessive development. Such development might entail designing layouts for data storage in text files so that data may be manipulated, and designing simple interfaces to access data stored in these files.

This functionality is readily available in relational databases. MySQL is an open source relational database, and, thus, has a significant cost advantage over other commercially available relational database management systems.

MySQL is highly scalable and is not difficult to administer. A trained database administrator is not required to manage MySQL installation, and management is relatively simple.

MySQL supports client APIs for a variety of programming languages (such as Perl, C and PHP), and client programs that access database information may be developed with such APIs. We are, therefore, provided more programming language choices for implementing the middle tier.

Oracle is also a very famous database package like My SQL but it is not open source. SQL Server is also not open source and developed by Microsoft. Any database package may be used to storing database.

Middle Tier

The middle tier produces run time HTML output by generating data in the middle tier itself or by retrieving data from the database. For example, in order to display a list of items offered on a site, the middle tier would return the required information from the database, manipulate it appropriately, then dynamically generate an HTML document that displays the items of interest.

Options for implementing the middle tier include:

CGI (Common Gateway Interface) programs written in Perl or C. These CGI programs access the database using the database language APIs.

Another way is Servlets on the middle tier. Servlets access the database by employing Java database APIs and the JDBC database driver. (JDBC drivers are available for most relational database management systems.)

Other options are Server-side scripting languages such as PHP or Perl. These languages support APIs for nearly all relational database management systems.

Client Tier

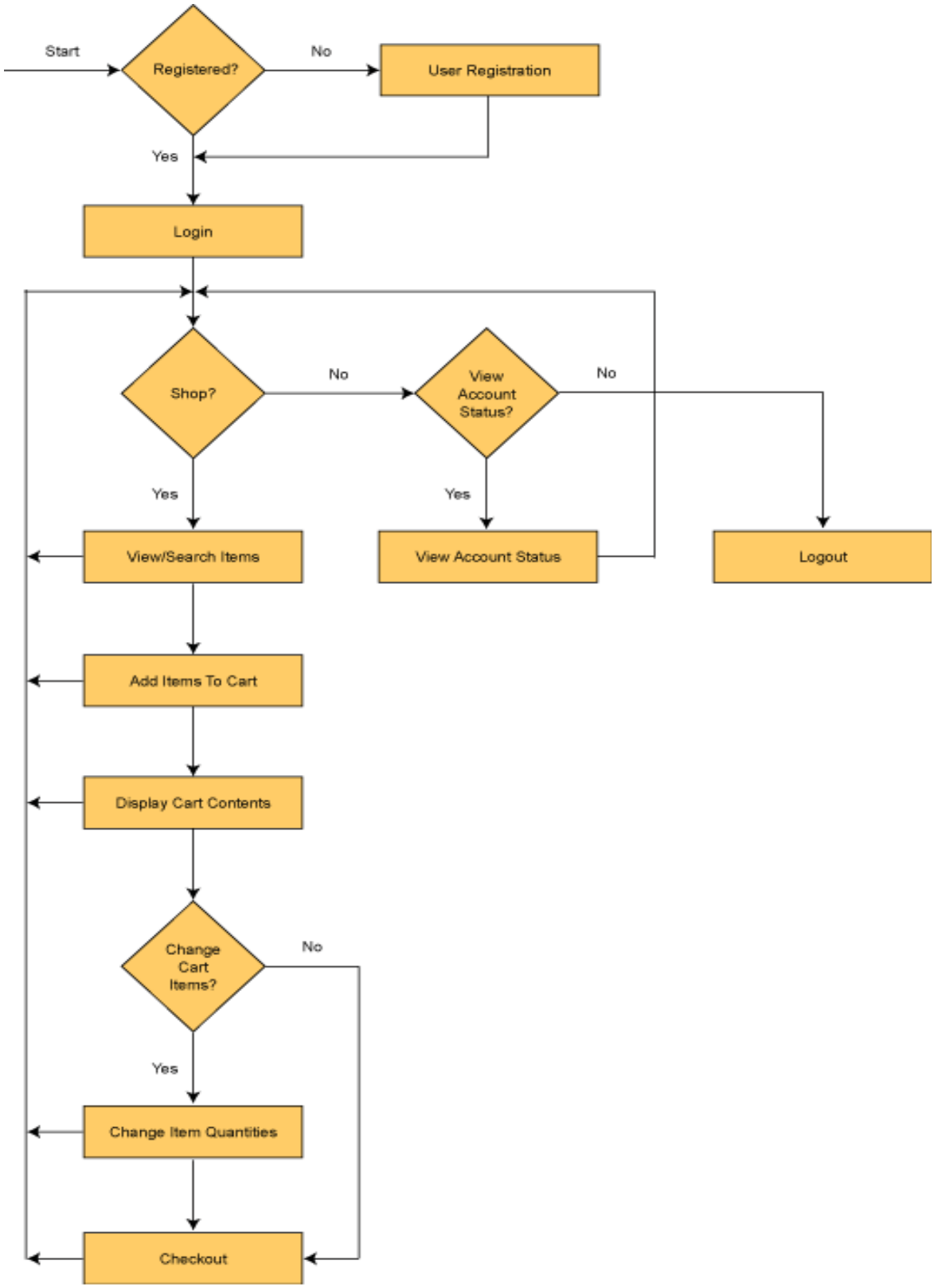
The user interface comes under client tier. The user interface is designed by the dynamic middle tier programs like PHP. So client gets web page as per his/her needs. For example, some client may want to buy LED TV with 32" size. So the page can filter to show only 32" size TVs to the client filtering others.

Consider the sequence of actions performed by typical Internet users for buying a product:

- New users arriving at an eCommerce site for the first time may create accounts. Existing users may authenticate themselves by providing their unique identifiers and passwords.
- After successful authentication, users may add items they are interested in purchasing to their shopping carts and view the list of items in their carts. (Of course, users may also wish to simply browse the electronic store without purchasing items.)
- When viewing shopping cart contents, users may wish to change the quantities of items they have selected.
- Site visitors may then checkout by confirming they would like to purchase the items in their shopping carts. The items in customers' shopping carts are then entered into the database. Users may then continue shopping or logout.
- Customers may also wish to return to the site at a later time to ascertain the shipping and delivery status of items they have purchased.

The flow chart below describes the typical interactions of site visitors with the shopping cart application. (Fig 2)

Fig 2



Buidling Business on the Net

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 irctc 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 over lapping, 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 fear 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 band width 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 storte 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 Cybersecurity

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

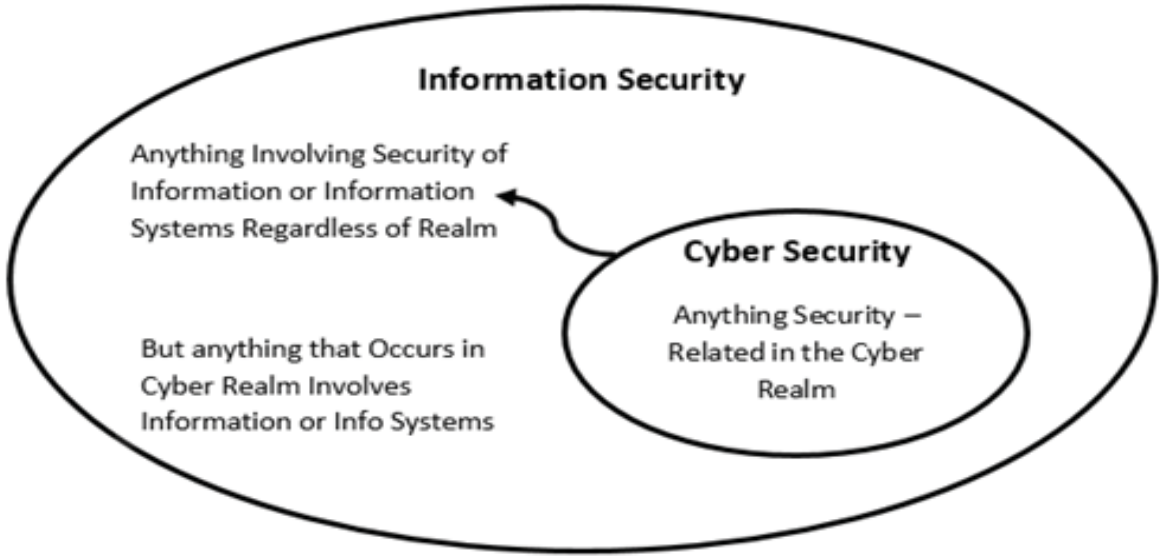
1 IT Security is assigned a low priority

- 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)

Fig 1



Relation between Information Security and Cybersecurity

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.

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 stakeholder 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.

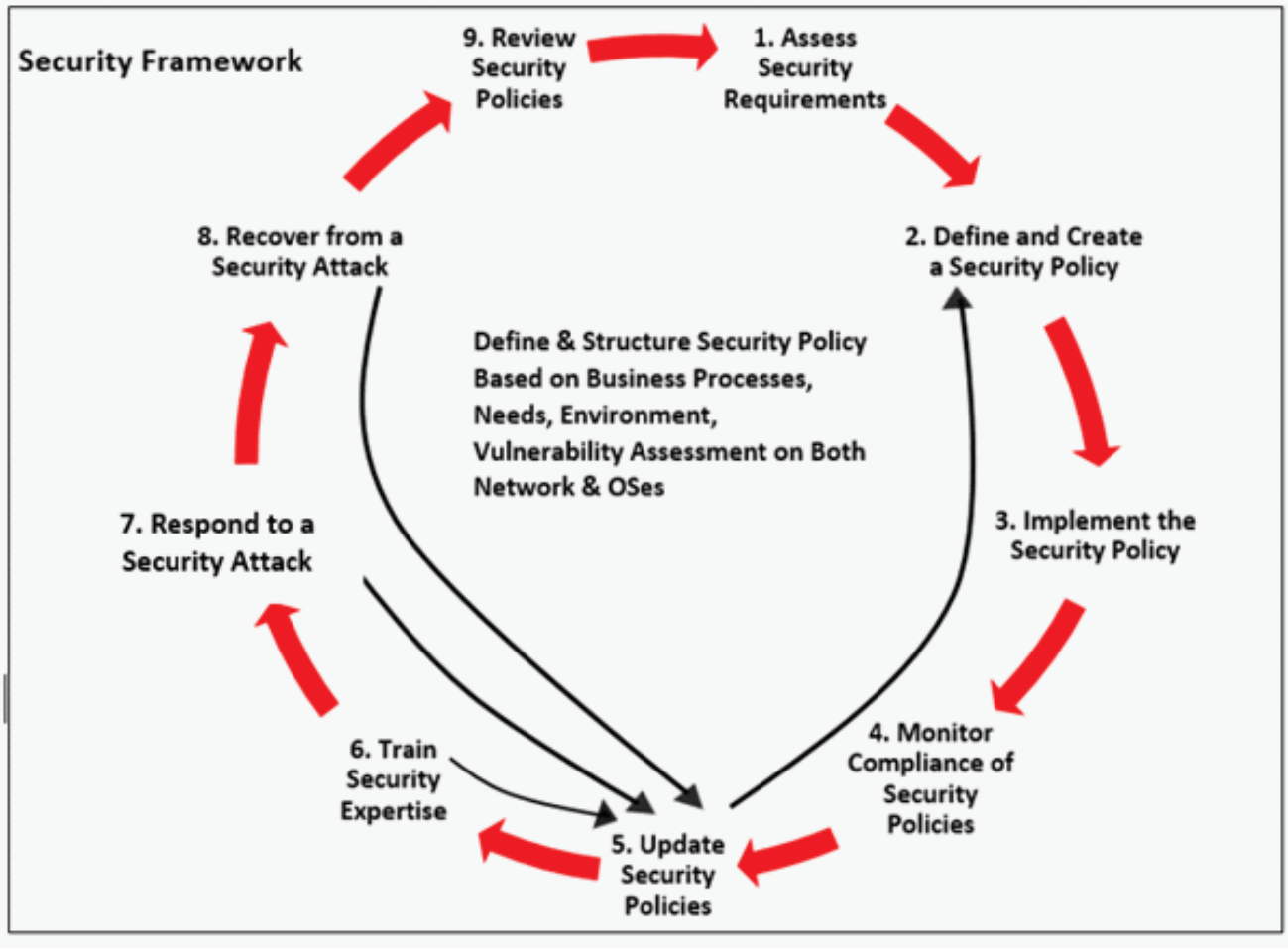
For an IT security system to work, a well-defined framework needs to be developed involving all stakeholders, and it needs to be updated over time to be useful. The information security frameworks facilitate the management process in considering the handling of data and implementation of system/ process 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, corrrective 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.

Identify	Protect	Detect	Respond	Recover
• Asset Management	• Access Control Events	• Anomalies and	• Response Planning	• Recovery Planning
• Business Environment	• Awareness and Training • Data Security	• Security Continuous Monitoring	• Communications	• Improvement
• Governance	• Information Protection and Procedures	• Detection Process	• Analysis	• Communications
• Risk	• Maintenance Assessment		• Mitigation	
• Risk Management Strategy	• Protective technology		• Improvements	

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.

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

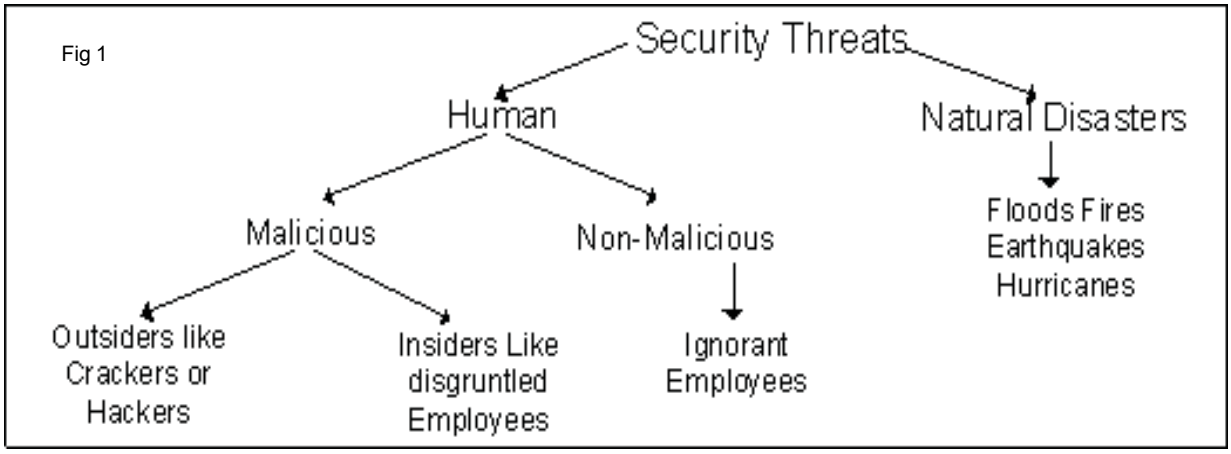
Once the threat has infected a device, it always aims to spread one way or another.

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

Fig 1



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 undergrounds 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 wants 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 (Sqli -> 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 based on 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.

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.

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 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 polices 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 Sandardization (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.

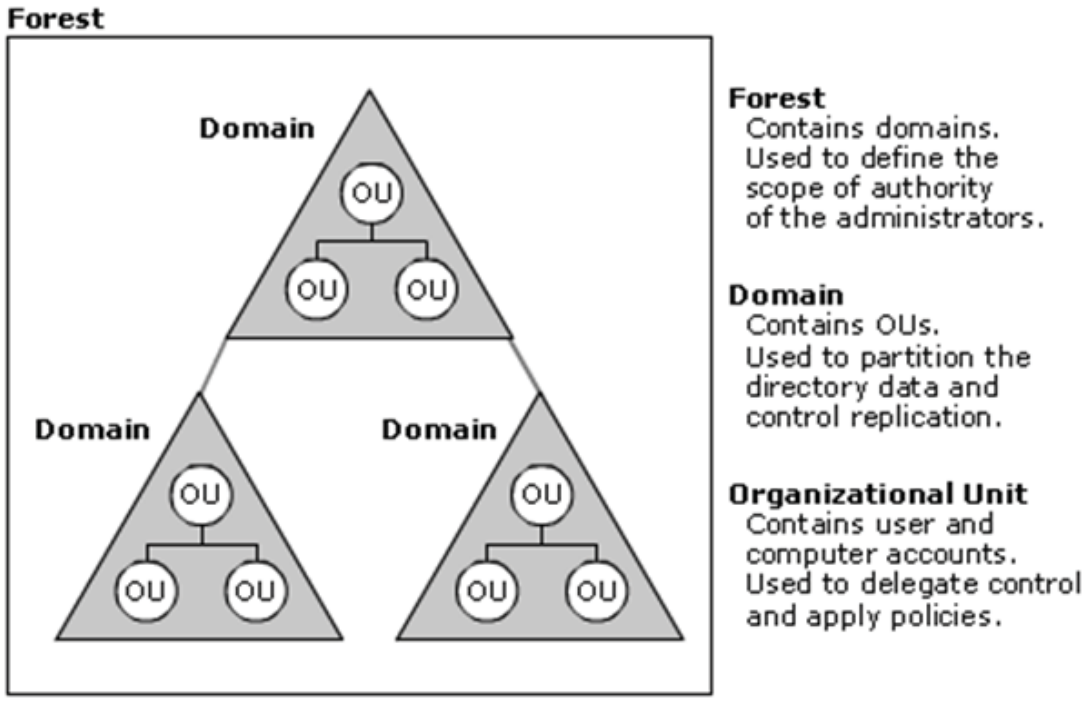
Within domains are organizational units. OUs can provide hierarchy to a domain, ease its administration, and can resemble the organization's structure in managerial or geographical terms. OUs can contain other OUs-domains are containers in this sense. The OU is the level at which administrative powers 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)

Fig 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 sub divided 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.

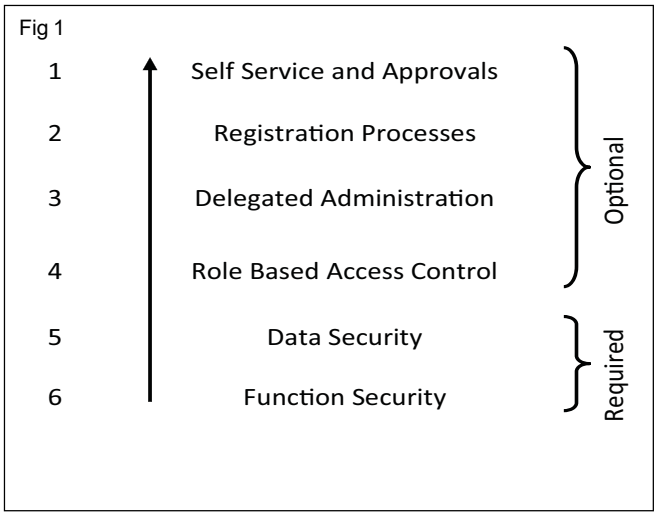
A security administrator maintains a database of authorizations based on the security policy of the organization. The reference monitor consults an authorization database in order to determine if the user attempting to do an operation is actually authorized to perform that operation. Auditing monitors 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:

Fig 1



- 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 with 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.

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	

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.

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 Countermeasures:

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, f 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 though 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 the 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 polices, 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 Reponses 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 situtation 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 prevalanet 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.