COMPUTERS AND COMMUNICATION TECHNOLOGY

PART II

Textbook for Class XI

राष्ट्रीय शैक्षिक अनुसंधान और प्रशिक्षण परिषद्
NATIONAL COUNCIL OF EDUCATIONAL RESEARCH AND TRAINING

Rs. 00.00

FOREWORD

The National Curriculum Framework (NCF) 2005 recommends that children's life at school must be linked to their life outside the school. This principle marks a departure from the legacy of bookish learning which continues to shape our system and causes a gap between the school, home and community. The syllabi and textbooks developed on the basis of NCF signify an attempt to implement this basic idea. They also attempt to discourage rote learning and the maintenance of sharp boundaries between different subject areas. We hope these measures will take us significantly further in the direction of a child-centred system of education outlined in the National Policy on Education (1986).

The success of this effort depends on the steps that school principals and teachers will take to encourage children to reflect on their own learning and to pursue imaginative activities and questions. We must recognise that, given space, time and freedom, children generate new knowledge by engaging with the information passed on to them by adults. Treating the prescribed textbook as the sole basis of examination is one of the key reasons why other resources and sites of learning are ignored. Inculcating creativity and initiative is possible if we perceive and treat children as participants in learning, not as receivers of a fixed body of knowledge. These aims imply considerable change in school routines and mode of functioning. Flexibility in the daily time-table is as necessary as rigour in implementing the annual calendar so that the required numbers of teaching days are actually devoted to teaching.

NCF–2005 envisages an increase in the optional number of subjects available for students of Classes XI and XII. This textbook marks an attempt in this direction. Its subject matter concerns how we work with computers, rather than about how computers works. It is expected that this textbook will be used with the help of methods of teaching which might enhance the flexible and participatory nature of new communication technology.

NCERT appreciate the hard work done by the Textbook Development Committee and its Chief Advisor, Professor M.M. Pant, Former Pro-Vice Chancellor,

Indira Gandhi National Open University, New Delhi. We are indebted to the institutions and organisations which have generously permitted us to draw upon their resources, material and personnel. We are especially grateful to the members of the National Monitoring Committee, appointed by the Department of Secondary and Higher Education, Ministry of Human Resource Development under the Chairpersonship of Professor Mrinal Miri and Professor G.P. Deshpande, for their valuable time and contribution.

As an organisation committed to systemic reform and continuous improvement in the quality of its products, NCERT welcomes comments and suggestions which will enable us to undertake further revision and refinement.

Director

New Delhi
May 2008

National Council of Educational
Research and Training

PREFACE

Computers have become a way of life in the present day world. That everyone should be educated about the implications of this technology is the need of the hour. Till now NCERT has not proposed a course or brought out a textbook in this area. The course outlined for this textbook is an attempt to design a generally friendly course that would not only be contemporary but also leave sufficient scope for expansion into unknown realms of computer activity that may emerge in future.

The National Curriculum Framework 2005 recommends that theoretical component of Higher Secondary stage should emphasise on problem solving methods and that the awareness of historical development of key concepts be judiciously integrated into the content of a subject. It also recommends that given the pervasive impact of Computer and Communication Technologies (CCT), a course related to this should address this infrastructure challenge seriously and explore viable and innovative alternatives with regard to hardware, software and connectivity technologies appropriate for rural and urban Indian schools.

NCF 2005 states that the book should act as an instrument for achieving social change in order to reduce the divide based on economic class, gender, caste, religion and region. NCF also speaks about tremendous effectiveness of the computer and computing technology in shaping modern society which has created the need for an educated public that can utilise such technology most effectively for the betterment of society and humankind. This book is aligned to these broad guidelines of NCF.

Presumably, this would be a coveted book for "everybody", irrespective of any particular stream of higher secondary stage, for it deals with the real challenges that the discipline is trying to solve. It emphasises development of problem-solving and, equally important, problem-formulating skills. It reduces the importance of technology and underscores the need for learning skills to manipulate the technology. It focuses on some of the real problems that come up with the expansion of the technology – security, piracy and digital identity. Above all, the course focuses as

much on the excitements as on the limitations of the new information technology. Effective integration of information and communication technologies into all curricula assists students in developing the abilities to use, manage, and understand the technologies.

The National Knowledge Commission refers to preparing a work-force adequately skilled and oriented to actively participate in the emerging Knowledge Economy. This book would be a major contributor towards developing the skills sets required for all school leavers.

This part of Class XI textbook contains six chapters under three themes/ units, viz., Web Publishing Technology; Teamwork and Web Based Collaboration Tools and Emerging Technologies. The book has been conscientiously designed and is the result of the renewed efforts of the Textbook Development Team comprising of school teachers, subject experts, academicians and technical experts from government, non-government and private institutions/organisations. Some of the members worked at the advisory level while others worked towards the actual developmental activity. It is hoped that the students will appreciate the beauty and logic of computers and communication technology. It has truly been a team work.

The course is not biased towards any stream; it can be opted as an elective subject with any other combination, be it science, commerce, arts or humanities. The students may or may not continue to study about computers beyond the higher secondary stage but it seems they will find the inherent logic behind CCT useful in any other branch they may like to pursue, be it administration, languages, social sciences, environment, engineering, technology, sciences, mathematics, medicine or any other branch of knowledge. In order to encourage learning through solving real life problem in collaborative environment, a full chapter is devoted to "Project based Learning". Also, chapter on "CCT Projects in Local Context" will provide insight about effective uses of CCT in improving quality of life of a common man, be a fisherman, a farmer, a teacher, or a businessman, a Government employee or anybody in their day today needs. Also with a hope that the child should be able to visualise that the journey from Abacus to PC has not come to an end, rather it is evolving continuously and efforts are going on bring further advancements in technology and its tools, a chapter on "Emerging Trends in CCT" has been introduced. For those who pursue computers beyond this stage the content of the book will certainly provide a sound base.

In this book we have tried to bring in a conceptual coherence. The pedagogy and the use of easily understandable language are at the core of our effort without sacrificing the rigour of the subject. The nature of the subject of CCT is such that a certain minimum use of mathematics is a must. We have tried to develop the mathematical formulations in a simple and logical fashion as far as possible.

This book has some features which, we earnestly hope will enhance its usefulness for the students. Each chapter is provided with objectives in

the beginning and a summary at its end for a quick overview of the contents of the chapter. There are some questions which require critical thinking which would make a student think about real-time application of CCT. Further, a large number of solved examples are included in the text in order to clarify the concepts and/or to illustrate the application of these concepts in everyday real-life situations. Some Practical activities/Case studies have been included which provokes students for deep thinking. Some of these are from real-life situations. Students are urged to solve these and in doing so, they may find them very educative. Some items inside boxes are introduced in many chapters either for this purpose or to highlight some special features of the contents requiring additional attention of the learners. Some information has been provided in shaded box which is meant for supplementary reading and not for evaluation. Glossary of terms and concepts is given towards the end which will act as a ready reckoner.

Completing this book has only been possible because of the spontaneous and continuous support of many people. We express our gratitude to the Director, NCERT, for entrusting us with the task of preparing this textbook as a part of national effort for improving general/school education. The Head, Department of Computer Education and Technological Aids, NCERT, was always willing to help us in our endeavour in every possible way.

The draft got excellent academic inputs from teachers, students and experts who sincerely suggested improvement during the development of this book. We are thankful to all those who conveyed these inputs to NCERT. We are also thankful to the members of the Review Workshop organised to discuss and refine the first draft.

We welcome suggestions and comments from our valued users, especially students and teachers. We wish our young readers a happy journey to the exciting realm of CCT.

M. M. Pant
Chief Advisor
Textbook Development Committee

THE CONSTITUTION OF INDIA

PREAMBLE

WE, THE PEOPLE OF INDIA, having solemnly resolved to constitute India into a **SOVEREIGN SOCIALIST SECULAR DEMOCRATIC REPUBLIC** and to secure to all its citizens :

JUSTICE, social, economic and political;

LIBERTY of thought, expression, belief, faith and worship;

EQUALITY of status and of opportunity; and to promote among them all

FRATERNITY assuring the dignity of the individual and the unity and integrity of the Nation;

IN OUR CONSTITUENT ASSEMBLY this twenty-sixth day of November, 1949, do **HEREBY ADOPT, ENACT AND GIVE TO OURSELVES THIS CONSTITUTION.**

TEXTBOOK DEVELOPMENT COMMITTEE

CHIEF ADVISOR

M.M. Pant, *Professor*, Former Pro-Vice Chancellor, Indira Gandhi National Open University (IGNOU), New Delhi

MEMBERS

Arpita Burman, *Scientist 'D'*, National Information Centre (NIC), Department of Information Technology, Ministry of Communication and Information Technology, New Delhi

Basav Roychoudhury, *Lecturer*, Rajiv Gandhi Indian Institute of Management (RGIIM), Shillong (Meghalaya)

C. Gurumurthy, *Director (Academic)*, Central Board of Secondary Education, 'Shiksha Kendra', New Delhi

Chandita Mukherjee, *Director*, Comet Media Foundation, Mumbai

Deepak Shudhalwar, *Lecturer*, PSSCIVE, Bhopal

Divya Jyoti, *PGT, Computer Science*, SLS DAV Public School, Mausam Vihar, Delhi

Gurpreet Kaur, *Head*, Department of Computer Science, G. D. Goenka Public School, Vasant Kunj, New Delhi

H.N.S. Rao, *Deputy Commissioner (Academic) (Retired)*, Navodaya Vidyalaya Samiti (NVS), New Delhi

Rajendra Tripathi, *Facilitator - People Development*, Azim Premji Foundation, Bangalore

M.P.S. Bhatia, *Assistant Professor*, Netaji Subhas Institute of Technology, New Delhi

Manish Kumar, *PGT*, Rajkiya Pratibha Vikas Vidyalaya, Raj Nivas Marg, Delhi

Mukesh Kumar, *Head*, Department of Computer Science, Delhi Public School, R.K. Puram, New Delhi

Prakash Khanale, *Vice Principal*, DEM College of Parbhani, Maharashtra

Rajni Jindal, *Assistant Professor,* Department of Computer Engineering, Delhi College of Engineering, Delhi

Sushila Madan, *Director-IT,* Vivekanand Institute of Professional Studies, *Shivaji Marg,* New Delhi

V.P. Chahal, *PGT Computer Science,* Jawahar Navodaya Vidyalaya, Mungeshpur, Delhi

Utpal Mallik, *Professor (Retired),* NCERT, New Delhi

Coordinator

Asha Jindal, *Reader,* DCETA, NCERT, New Delhi

Team Member

Rajaram S. Sharma, *Professor* and *Head,* DCETA, NCERT, New Delhi

ACKNOWLEDGEMENT

CONSTITUTION OF INDIA

Part III (Articles 12 – 35)
(Subject to certain conditions, some exceptions
and reasonable restrictions)
guarantees these

Fundamental Rights

Right to Equality

- before law and equal protection of laws;
- irrespective of religion, race, caste, sex or place of birth;
- of opportunity in public employment;
- by abolition of untouchability and titles.

Right to Freedom

- of expression, assembly, association, movement, residence and profession;
- of certain protections in respect of conviction for offences;
- of protection of life and personal liberty;
- of free and compulsory education for children between the age of six and fourteen years;
- of protection against arrest and detention in certain cases.

Right against Exploitation

- for prohibition of traffic in human beings and forced labour;
- for prohibition of employment of children in hazardous jobs.

Right to Freedom of Religion

- freedom of conscience and free profession, practice and propagation of religion;
- freedom to manage religious affairs;
- freedom as to payment of taxes for promotion of any particular religion;
- freedom as to attendance at religious instruction or religious worship in educational institutions wholly maintained by the State.

Cultural and Educational Rights

- for protection of interests of minorities to conserve their language, script and culture;
- for minorities to establish and administer educational institutions of their choice.

Right to Constitutional Remedies

- by issuance of directions or orders or writs by the Supreme Court and High Courts for enforcement of these Fundamental Rights.

CONTENTS

CONTENTS OF PART-I

Web Page Designing Using HTML

9

Objectives

After completing this Chapter, the student will be able to:

- *recognise HTML tags and their attributes,*

- *use HTML tags and attributes to develop web pages,*

- *use HTML tags and attributes to format text and images, design layouts, present tables and lists and create hyperlinks,*

- *use HTML forms to add interactivity to web pages, and*

- *design a web site and construct it using HTML.*

"Logic will get you from A to B. Imagination will take you everywhere."

Albert Einstein

Introduction

As we have learnt earlier, the Internet is a very large network of computers spread across the world. The Internet allows for each of these computers to send and receive information from each other. One of the major applications of the Internet is the World Wide Web (WWW). The World Wide Web or the web in short, can be thought of as an interconnected set of documents, images, audio, video files or software files.

When you connect to the Internet and look around (we call it browsing), you are using a very interesting feature of the web, hyperlinks. Each time you click on a highlighted piece of text or image, you jump to another piece of text or image and this could be on the same page, another page on the same hard disk or on a page residing on another computer, half way across the world.

How this works and how we could make it happen is what we will learn in this chapter. Web pages use a language called the Hyper Text Markup Language (HTML). The browser

applications (Microsoft's Internet Explorer, Mozilla's Firefox, etc.) are designed to interpret HTML to display text, images and using hyperlinks, navigate the web.

9.1 INTRODUCTION TO HTML

Let us begin by looking at a web page (Figure 9.1) and analysing its features. This web page has the following major sections :
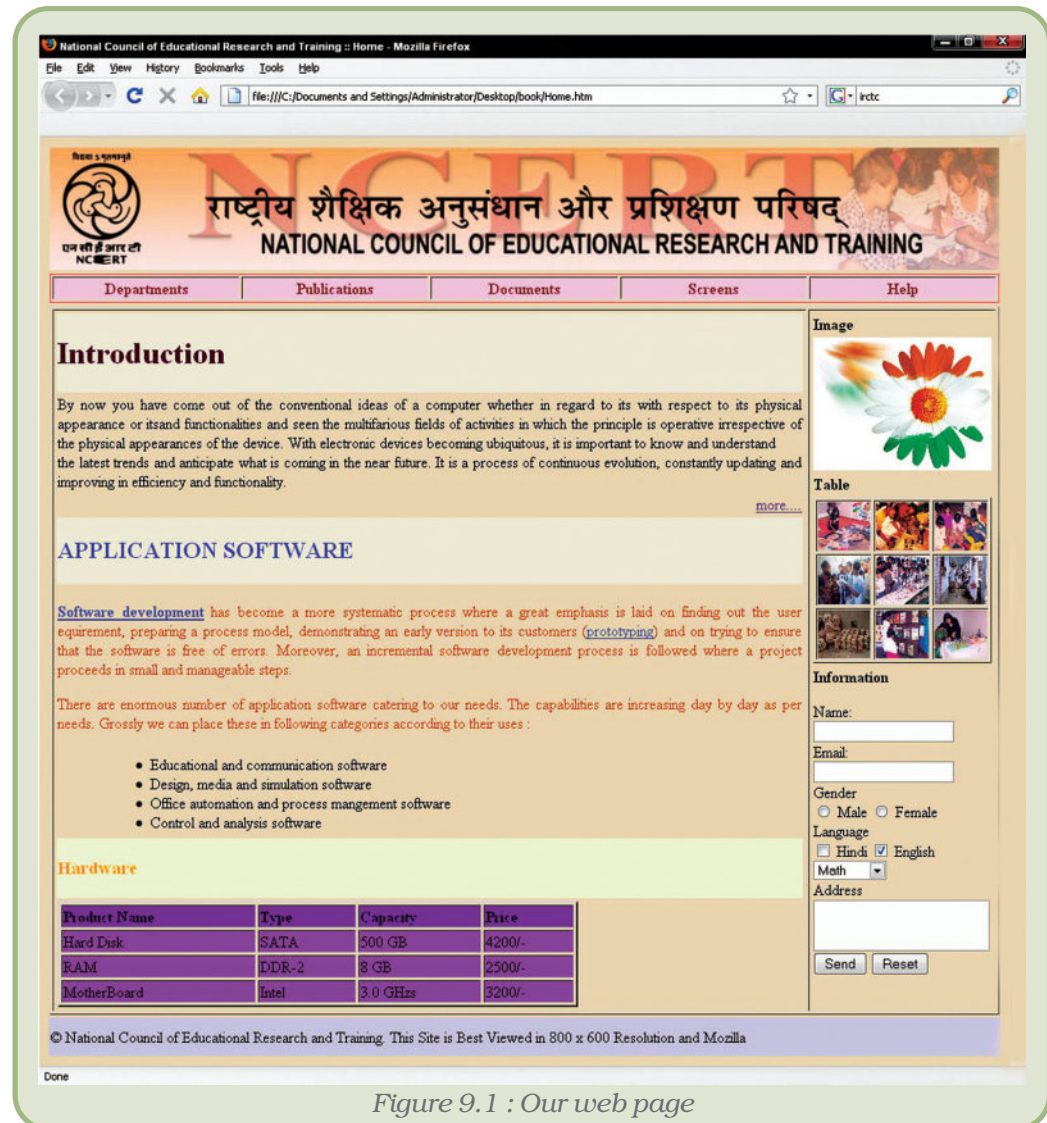
Figure 9.1 : Our web page

- **a header area** : In our example, it consists of a mast head, which signifies the subject of the web page. It may also contain a picture, a logo, etc., depending on the purpose of the page.

- **a menu bar** : This is a set of links, designed as buttons. Each link takes you to a separate section of the web page. In a way, this is like the chapter headings in a book.

- **body of the page** : This part of the page contains all the content that you wish to present. The content could be of a variety of types – text, images, tables, forms, etc. We have organised the page in two columns. While the left column contains text, formatted in different ways, the right column contains some features unique to a web page – links, images which link to an image gallery, a comment form and a poll. It is such features which make a page interactive.

- **a footer area** : This is at the bottom of the page and usually contains disclaimers, copyright information, etc. In our case, the footer text states, "All material on this site can be freely downloaded, distributed or reused".

We will analyse the page and its sections further. We will also learn how to construct such web pages. In order to extend your knowledge, do look at a variety of web pages, each time noting down the layout and categorisation of content. Web page design has evolved into a very sophisticated activity, where artists and programmers work together to put together very elaborate features in order to make web pages interactive.

As we mentioned in the introduction, you need an application software called an Internet browser to view web pages. While all browsers are designed to interpret HTML and display web pages, they can be designed for different operating systems. You can have many browsers installed on your computer and use them simultaneously. Most browsers are free and can be downloaded from the web. Some popular browsers are listed below :

Browser	Operating System
Mozilla Firefox	Microsoft Windows, Linux or Apple's MacOS
Internet Explorer	Microsoft Windows
Google's Chrome	Microsoft Windows
Opera	Microsoft Windows
Safari	Microsoft Windows and Apple's MacOS

During the course of this chapter, you will encounter a number of browser commands. It would be unwieldy to refer to the commands in each kind of browser. Therefore, we will use Mozilla's Firefox and refer to its commands. Every browser will have commands which will achieve similar purposes. They may however be placed in a different category, or called something else. So, when we mention a command, do check out how you will do a similar thing in Internet Explorer, Chrome or Opera.

Every HTML file (also referred to as a web page) is a text file. It does not contain images, sound, or such information. The HTML file has

references to such files. So when you see a photograph on the web page, the HTML file has called the image file and displayed it in the appropriate place on the web page. The fact that this is a text file, makes it platform independent – meaning that the same web page could be seen on a Windows, Unix, Linux or a MacOS computer. This is also the reason why the web is emerging as a universal medium of communication.

We mentioned that what traverses the web is the HTML code. It is this code that helps a browser construct the web page. But, how do you see the HTML code? Right-click on the page and from the options displayed, select view page source. You should now see the HTML code. This may look very threatening, but let us assure you that it is actually quite simple and you should be able to write it by the end of this chapter.

9.2 HTML TAGS

What does your HTML code contain? It contains the contents of the web page, information about how and where to display each of the content elements, and if you input information or click on a link, what actions to take. In other words, a HTML page contains the content, the programming and the design elements.

The basic element of the HTML code is a tag. Let us look at the code of our web page (Figure 9.2).
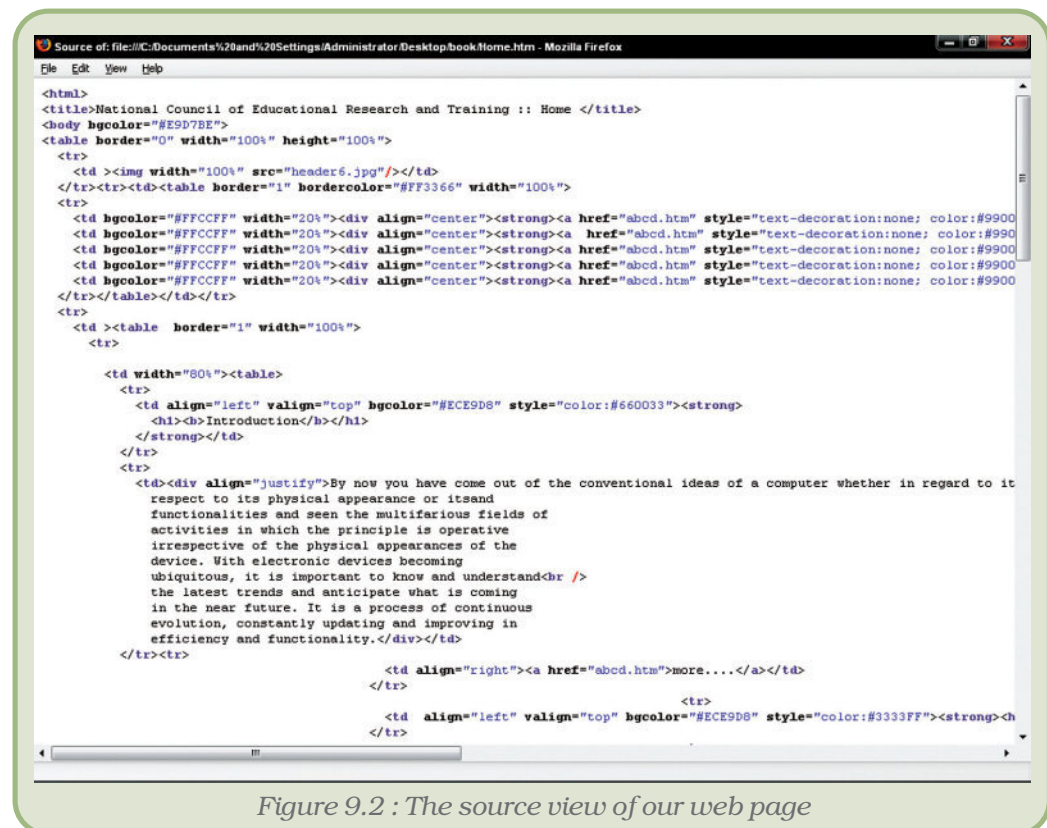
Figure 9.2 : The source view of our web page

Notice the code <HTML> at the top of the page. If we scroll down to the bottom of the page, we will see the code </HTML>. We refer to such codes as **tags**. Learning to make web pages is all about learning to use these tags to present content the way you want. The general format for a HTML tag is :

```
<tag_name> affected text </tag_name>
```

The opening tag consists of just the tag name and it instructs the web browser to start tagging (formatting) the text. The ending tag includes a slash preceding the tag name. This "/" slash tells a web browser to stop tagging the text. Let us look at a few examples:

If you want your text to be highlighted, say using bold type face, you use

```
<b> HTML tags </b>
```

and what you get on the web page is **HTML tags**. Notice that the tag is not case sensitive, that is, <B> HTML tags </B> would have yielded the same result. An interesting aspect of a HTML tag is that if the browser does not know what to do with a given tag, it will simply ignore it!

There are two types of tags :

• Container tags : The HTML tags which require a closing tag are called container tags. These tags are also called *paired* or *wrapper* tags. The text is one such pair. If you do not close such tags, all the following text would be treated as if they are part of this tag.

• Non-container tags : The HTML tags which do not require a closing tag are called non-container tags. These tags are also called *stand alone or empty tags*. A typical example is that of <br>. This tag inserts a line break.

9.2.1 ATTRIBUTES

Some tags are complete in themselves. In other words, they have only one option. For instance, when you want your text, say italicised, you use <i> text </i>. But if you wish to display text, say in some particular font, how will the tag look then?

```
<font face="Arial"> text</font>
```

Notice the tag in this example is . What about the remaining part, face= "Arial" We refer to this additional information as the tag's attribute. So, in our example, the tag is font, its attribute is face and the value given to the attribute is Arial.

Attributes may have many different values you may like to use simultaneously. For instance, you may not only like a particular font,

but also a particular size of letter and in a particular colour. How do you do this? The tag would now be

```
<font face="Arial"; size=14px; color=#ff0000">text </font>
```

Note the use of quotes to indicate values and semi-colons to separate attributes. The general form of the tag is :

```
<tag_name attribute-name = "attribute-value">....</ tag_name>
```

As a general rule, most attributes – that include only letters, digits, hyphens, or periods – work fine without quotes. Attributes that have other characters, such as spaces, % signs, or # signs, however, *always* require quotes. Nevertheless, it is a good habit to enclose all attribute values in quotes.

The purpose of the attribute is to extend the scope of a tag. It provides additional information about the tag, to the browser.

How does the browser treat a tag, which requires an attribute, but no attribute is mentioned. For instance in our example if we had just used text, how would the browser show it. The best way to figure this out is to try it out yourself.

Each browser has a default setting, which is the value it takes when no other value is defined. In Windows Internet Explorer, the default font setting is "font:Times New Roman; font-size:12pt and color:#000000. So if you only have the text tag without any attributes, then the text would be in 12 point, Times New Roman font and its colour would be black. You can of course change this default setting to whatever you wish. Go to Tools->Internet Options and click on the font button under the general tab. There are many other defaults and you can set any or all of them to customise your browser to your requirement.

9.3 STRUCTURE OF A HTML DOCUMENT

In the subsequent sections, we will analyse the web page we have made (Figure 9.1) and in the process learn about various HTML tags. Let us look at the HTML coding of our web page. Right-click on the page and opt for view page source. (Figure 9.2)

A basic concept of programming is to present the code in a very structured format. Not only does it make it easy to read, but also helps detect and correct errors. In HTML, we do these in the following way.

The entire HTML document is enveloped between a pair of tags: <HTML> and </HTML>. In other words, when enclosed between these two tags, the browser treats the page as a web page.

The sections of the page are then defined. In our page we have a header, a body and a footer. The tags are <head></head> and <body></body>.

236

So the HTML page looks like this :

```
<html>
<head>
Instructions for the browser goes here;
</head>
<body>
Everything on the web page goes here;
</body>
</html>
```

All the contents of the web page displayed in the browser window are between the tags <body></body>. In our web page, this includes the following :

• the banner, Computers and Communication Technology against a coloured background;
• the menu bar with buttons to each chapter;
• all the text in the left column, again against a coloured background;
• a set of hyperlinks;
• two input boxes; and
• the footer

We will explore the tags associated with each of these in the subsequent sections.

9.4 STARTING WITH CODING

What tools do we need?

To create a HTML document, we need two basic tools:

1. **A HTML editor**, to create and save the HTML documents. Notepad or Word Pad for Windows, vi for Unix, and Teach Text for Macintosh are good examples of text editors. There are more advanced and feature rich HTML editors, but a text editor will suffice for our purpose.

Note : Do not use a word processor like OpenOffice.org Writer or Microsoft Word. These applications generate a lot of invisible code for formatting, which is beyond the capability of a browser. But if you have to use it, save the document as plain ASCII text. In this format, all additional code generated by the word processor is removed.

2. **A Web browser**, to view and test the HTML documents. The most common and popular web browsers are Mozilla Firefox and Internet Explorer.

Creating Our First HTML Document

We will now create the web page that we explored in figure 9.3. We will use Windows Notepad as the text editor. Type in the basic tags. Save the file in a convenient location with a suitable name, say **home**.

237

Notepad will give the file a default extension of **.txt**. But we wish to make a web page. Therefore name the file **home.htm**
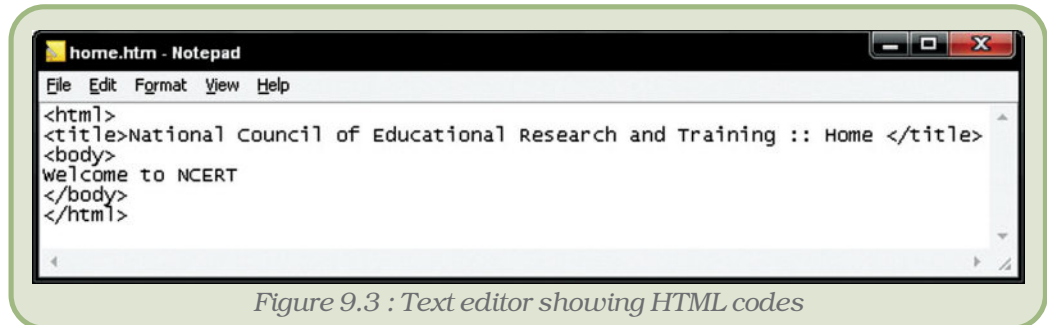
Figure 9.3 : Text editor showing HTML codes

Let us now preview our web page. Open the file home.htm in a browser (Figure 9.4).
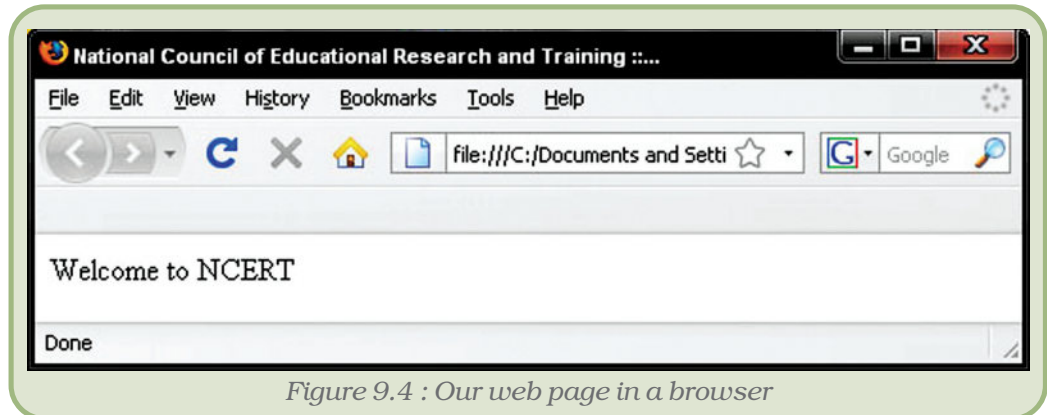
Figure 9.4 : Our web page in a browser

We can continue to edit the file home.htm, adding additional HTML code. Each time you wish to preview your work in the browser, refresh your browser (go to View ->Reload in the Mozilla Firefox browser or press Ctrl+R). The browser reloads the page showing you the latest version.

Adding the Header Section

Open the file home.htm in Notepad. Insert the tag pair <head></head> after the <html> tag and before the <body> tag. The code that goes between the <head></head> tags are not displayed by the browser, with the exception of the title tag. This part of the html document contains all references to scripts, special media players, special fonts, particularly when the web page is made for some other language, the meta data, which is used by search engines to index this web page, and any styles that are used for the content, etc. In our example, we have no such special requirement. So the header section of our web page contains only the <title>text for the title</title> tags. Insert this tag and save the page. Preview the web page in your browser.

What we write between the tags <title></title> is displayed on the top of the browser window, above its menu bar. Notice the text "National Council of Educational Research and Training :: Home". This will also appear when you print the web page (Figure 9.5).

238

Figure 9.5 : The title of our web page

9.5 CONTENTS OF THE WEB PAGE – FILLING BETWEEN THE <BODY></BODY> TAG

9.5.1 BEAUTIFYING THE WEB PAGE – THE "BGCOLOR=" ATTRIBUTE

The body tag can also take a number of attributes. We will look at only one of them, namely "bgcolor=". This attribute is used to apply a background colour to the web page. Let us see a few examples:

```
<body bgcolor="cyan"> text </body>
<body bgcolor="#00ff00"> text </body>
<body bgcolor="red" background="bg.jpg"> text </body>
```

From the first two examples, you will notice that you can either use the name of the colour or its hexadecimal code. You may look up the hex codes in a graphics application like Gimp or Photoshop, or look up on the web.

From the third example you will notice that we have defined not only a colour, but also an image for the background. The image will be overlaid on the coloured background. So if the image is big enough, you will only see the image as the background. If it is smaller than the screen, the background colour will be visible all around. You will learn about managing images in a later section.

In the previous section, we examined the <head></head> pair of the HTML tags, included a title for our web page. Let us continue our exploration of our web page. Our web page could be visualised as figure 9.6 consisting of the following:
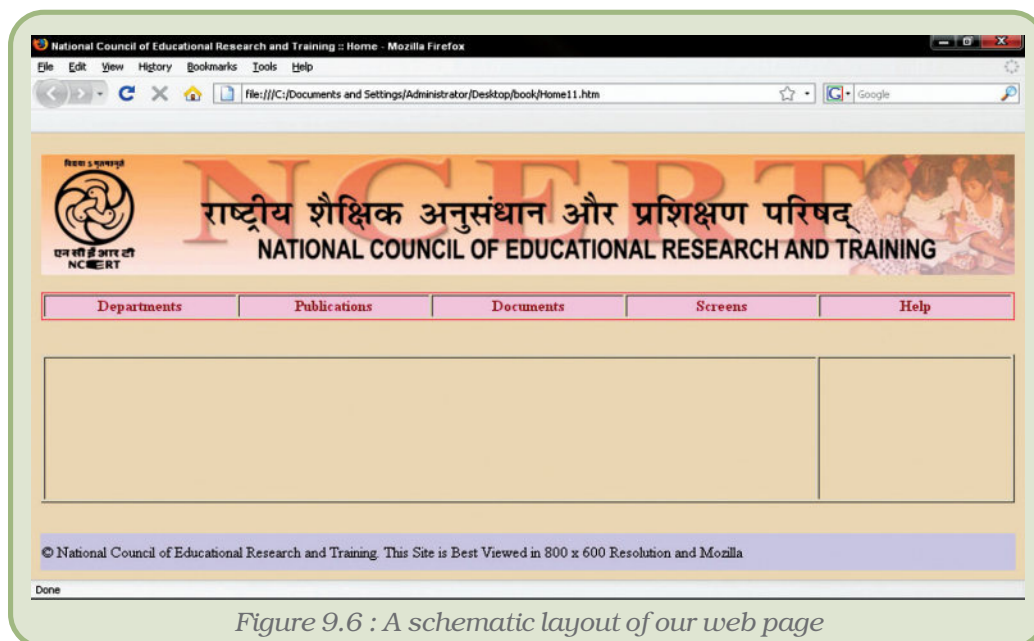
Figure 9.6 : A schematic layout of our web page

239

9.5.2 THE COMMENTS TAG: USE OF <!— —>

When you write a long piece of coding, or if you are writing a piece of code which you want someone else to edit or use, it would be a good habit to mark out sections with labels. At the same time, you do not want these to be visible on the web page. One useful tag for this purpose is the comment tag <!— text —>. For example,

<!— the following section will display the image gallery —>

Anything in between the comment tags is ignored and is **NOT** displayed in a browser. Comments can be multi-line too, and only one set of start and end tags can be used to enclose all the lines.

9.5.3 THE BANNER AND THE USE OF THE <HN></HN> TAG

Notice that the banner has text in a large font size. There are many ways of achieving this. You could define a font size, you could use pre-defined sizes, or even use an image containing the text. The last option is preferred when a particularly unique style of text or text effect is to be used, something which cannot be obtained using normal HTML coding.

While it is possible to define the exact font size (we will look at this option a little later), HTML provides six levels of text for use as headings. They are numbered 1 to 6, with 1 being the largest and 6 being the smallest. But, how big is h1 and how small is h6? These are relative sizes. Now, if you have six sentences in sizes h1 to h6 as in figure 9.7 and go to view-> zoom->zoom in or press Ctrl++, each of these sentences become bigger. Continue doing this and the text keeps growing in size. They maintain their relative sizes. Now, go to view ! zoom ! zoom out or press Ctrl—, the text becomes progressively smaller, still retaining their relative sizes.

The tags used in this case are <h1>text</h1>, <h2>text</h2>, etc. all the way to <h6>text</h6>. Look at these examples.

Introduction

APPLICATION SOFTWARE

Hardware
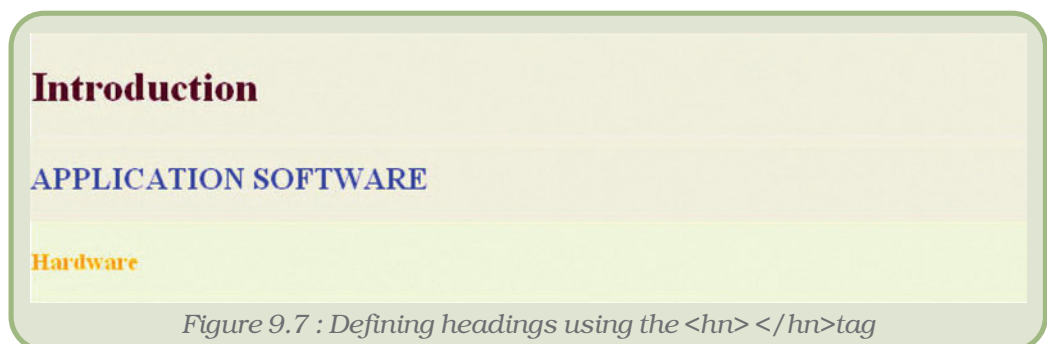
Figure 9.7 : Defining headings using the <hn> </hn>tag

9.5.4 THE USE OF THE <FONT></FONT> TAG

The <hn></hn> tag of course will use the default font of the browser and display text in varying sizes. So, what do you do if you want the text to be displayed in different fonts?

If you use a word processor like OpenOffice.org Writer or Microsoft Word, you are already familiar with the use of different font faces. These are different styles of lettering. To achieve this in HTML, we use the text tag. If you do not use any attributes, then the browser chooses its default font face. But if you use the attribute font-face= or simply face= and give it a value, the name of a font, then the text would be displayed using this font. Of course, the computer on which the web page is viewed should have that font installed.

The HTML code reads:

text to be displayed

You may also like to use other attributes like size= or color= and put in values of your choice. With size, you may choose +1, +2, etc. -1, -2, etc. or use absolute values like 24px, where px stands for pixels or even use percentages, e.g., size=30%. With colour, first of all remember to use the American spelling "color" and then use either the name or the hexadecimal (or hex code) of the colour.

You will notice each paragraph of text is in a different font face, font size and colour. The HTML code for the page can be seen in figure 9.8.

Figure 9.8 : The use of the tag

9.5.5 TAGS FOR HIGHLIGHTING TEXT

While there are a large number of font faces, in a variety of languages, following different styles of calligraphy, many of them are not suitable for reading large chunks of text. Also, use of too many different font faces on the same page can be strenuous for the reader. One would therefore prefer other ways of highlighting.

We already encountered one such technique when we used the <head> tag. You can use text of different sizes to highlight. While this may be suitable for titles and headings, how do we highlight words or phrases within running text. Usually, we use bold, italics or underline. Let us look at the tags which can do this.

This is how we convert text into bold;	Display the results here **bold**
This is how we <i>italicize</i>text; and	Display the results here *italicize*
This is how we <u>underline</u> text.	Display the results here <u>underline</u>

You can also achieve bold face text using text and italicized text using text. Do remember to use the closing tag. If you don't, all the subsequent text will be highlighted. Try this out on your web page.

9.5.6 MAKING PARAGRAPHS AND SEPARATING CHUNKS OF TEXT

Running text is usually broken into paragraphs, and with the advent of word processors, text is also justified (aligned) in different ways. We may also have to force a line break to avoid an orphaned or a widowed line (a single line of text on the bottom or top of a page). Let us explore the tags used for this purpose.

The <p>text</p> tag :

Look at the text in our web page home.htm again. The first paragraph is titled Introduction. Look at the structure of the paragraph (see Figure 9.9). After the title you have a blank line, followed by a block of text. The next paragraph doesn't begin immediately. You have one or more blank lines in between.

Now examine the tags which will help us achieve this. The tag pair <p></p> is used to define a paragraph. The initial or opening tag <p> will define a blank line before the tag. So the paragraph enclosed by these tags will separate it from the previous paragraph. But, the end or closing tag </p> will not define a line after the tag. So the next piece of text will have to begin with <p>.

When used on its own, this tag will define blank lines. This is useful when you wish to forcibly separate blocks of text. See the example below.

Figure 9.9 : This piece of text will have a blank line in between

The <br> tag :

You will notice that the <p></p> tag pair provides a large blank between the paragraphs. When you use the tag to provide blank lines, it may result in the page looking ugly. You will prefer to provide smaller gaps. Also, you may like to forcibly close a line and want to continue on the next line. When you type in text, depending on the size of your window, you may notice the text shifting to the next line. But if you resize the window, the text rearranges. This means that if the window were very big, all the text would come on to the same line. Without a line break defined, the text will not be forced on to the next line.

The tag to use in these contexts is <br>, which stands for a line break. This tag moves the following text into the next line. It is a standalone tag without a closing tag.

> **Note :** The line break tags <br> can be used to give as many blank lines as desired but the <p> tag can't be used more than once. In other words, specifying <p><p><p> won't give us 3 empty lines, it will just give us 1 empty line. But specifying

 gives us 3 empty lines.

The <hr> tag :

If you wish to separate two sections of your web page using a line, you can do it in two ways. One, you use an image. Two, you use the <hr> tag. This again is a standalone tag and stands for horizontal rule. You may use various attributes with this tag, as shown in the examples below.

```
<hr width=60>, <hr width=70%>, <hr size=5>
```

While width indicates the length of the line, size indicates the thickness.

9.5.7 ALIGNING TEXT

Most times text is aligned with the left margin of the page. In particular cases, you may also like to right align it. If it is a heading, you may also like to center it on the page. Let us look at the paragraphs in our web page. The first paragraph is left aligned. Notice that while each line starts from the same vertical position on the left, they stop at different places on the right. The align="left" attribute can be used with a variety of tags. For instance, with the heading tag, or the paragraph tag or even within tables.

243

For example <p align="left">This text is left aligned. </p>

Similarly, you can use the align="right", align="center" and align="justify" attributes to align the text to the right, center or to justify – align the text to both the left and right margins. This attribute can also be combined with various other attributes like font, colour or size.

9.5.8 PRESENTING LISTS OF CONTENT

There are two commonly used types of lists – a bulleted list or an unordered list, and a numbered list or an ordered list.

An example of an Unordered List...	
• bicycle • bullock cart • car • bus • train • ship	``` bicycle bullock cart car bus train ship ```

The tag pair <ul> and stands for the unordered list, while stands for list item. You may use a to close the tag, but this is optional. Notice the bullet before each item. This is generated by the <li> tag. There is no limit to the number of list items you may have in a single list.

The ordered list is very much similar in structure to the unordered list, except that the opening tag for the list is <ol> instead of <ul>, and the closing tag is instead of . List items within the list still use the same tags.

An example of an Ordered List...	
1 Argentina 2 Bhutan 3 Canada 4 Denmark 5 Ethiopia 6 France	``` Argentina Bhutan Canada Denmark Ethiopia France ```

The Ordered list offers the following three attributes :

The **TYPE** attribute of the <OL> tag can be used to change the type of numbering displayed in front of the list items (1/A/a/I/i). For example :

<OL TYPE=1> makes numbers (1, 2, 3) (this is the default)

<OL TYPE=A> makes uppercase letters (A, B, C)

<OL TYPE=a> makes lowercase letters (a, b, c)

<OL TYPE=I> makes uppercase Roman numerals (I, II, III)

<OL TYPE=i> makes lowercase Roman numerals (i, ii, iii)

The **START** attribute of the <OL> tag is used to start a numbered list with a value other than "1" (or A, a, I, or i). For example :

<OL TYPE=1 START=5> will start the numbering from 5

<OL TYPE=A START=5> will start the numbering from E

The **VALUE** attribute of the <LI> tag is used to change specific numbers within a list. For example :

<LI TYPE=1 VALUE=5> will display the number 5.

9.5.9 TABLES – THE USE OF <TABLE> </TABLE> TAG

Tables are a powerful technique of organising content on a web page. There are two significant ways in which the concept of tables are used; one where the table is visible on the web page and two where it is invisible, but restricts content into particular regions of the web page. We will consider examples of both these ways.

Let us examine the structure of a typical table. Look at figure 9.10 below.


```
<table width="100%" >
    <tr  valign="top" ><td valign="top" ><b>Image</b></td></tr>
    <tr><td><img   width="100%"src="flower.jpg "/></td></tr>
    <tr ><td><b>Table</b></td></tr>
    <tr><td>
            <table border="1" width="100%">
                <tr>
                    <td width="20%"  height="50" >
                    <img src="1.jpg" height="50" width="100%" />
                    </td>
                    <td width="20%"  height="50"><img src="2.jpg" height="50" width="100%" /></td>
                    <td width="20%"  height="50"><img src="3.jpg" height="50" width="100%" /></td>
                </tr>
                <tr>
                    <td width="20%"   height="50"><img src="4.jpg" height="50" width="100%" /></td>
                    <td width="20%"  height="50"><img src="5.jpg" height="50" width="100%" /></td>
                    <td width="20%"  height="50"><img src="6.jpg" height="50" width="100%" /></td>
                </tr>
                <tr>
                    <td width="20%"   height="50"><img src="7.jpg" height="50" width="100%" /></td>
                    <td width="20%"  height="50"><img src="8.jpg" height="50" width="100%" /></td>
                    <td width="20%"  height="50"><img src="9.jpg" height="50" width="100%" /></td>
                </tr>
            </table>
    </td></tr>
    <tr>
        <td><strong>Information</strong><br>
        <FORM action="ncert.htm" method="post">
        <P>
        <LABEL for="firstname">Name: </LABEL><br>
        <INPUT type="text" id="firstname"><BR>
        <LABEL for="email">Email: </LABEL><br>
        <input name="text" type="text" id="email">
        Gender<BR><INPUT type="radio" name="sex" value="Male">
        Male<INPUT type="radio" name="sex" value="Female">
        Female<br>Language <br><input type="checkbox" name="option1" value="Milk">
        Hindi<input type="checkbox" name="option2" value="Butter" checked>
        English<BR><INPUT type="submit" value="Send">
        <INPUT type="reset">
        </p>
            </FORM>
    </td></tr>
            </table>
    </td></tr>
</table>
```
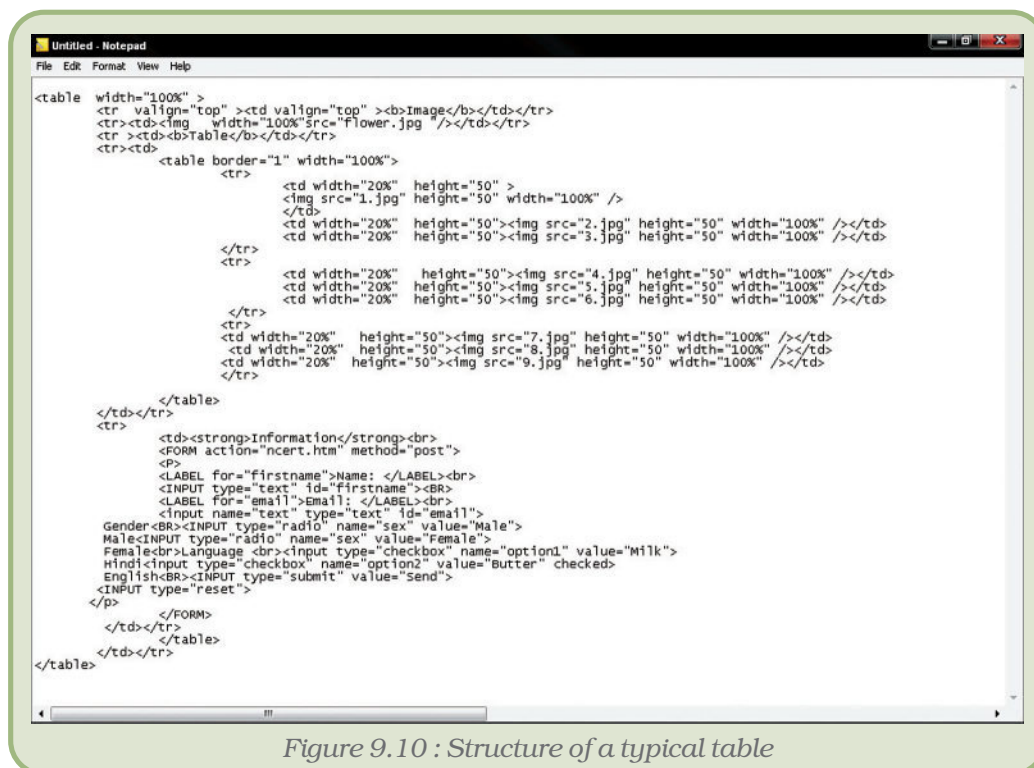
Figure 9.10 : Structure of a typical table

The table is organised in to rows and columns. Some rows and columns can be merged to form larger spaces within the table. They may also be split in to multiple rows or columns. The table may have a

header row, indicating what each column contains. Alternately it may have captions in the first column, indicating what each row contains. Each of these rows, columns, individual cells or captions may be highlighted in different ways, using different font faces, sizes of letters, emphasising text, aligning it differently, or filling the background with a colour or image. So there can be an enormous variety of tables.

The rows and columns of tables are demarcated by lines. In the HTML case, we refer to them as borders. Their colours or their line thickness could be modified. The text inside the cells are separated from the borders. The cells themselves can be separated from each other. In effect, when we learn to use the tag pair <table> </table> we are learning a whole new way of organising content.

This large variety of presentation techniques can be achieved using a very small number of tags, tag attributes and values.

The <table> </table> pair is the defining pair. All tags, attributes or values, text or images form the content of the table. A simple three row, two column table. The first row contains captions for each column. How does the HTML code look like? See figure 9.11.

```
<table border="1">
<caption align="center" >This is our table's caption</caption>
<tr><td>first cell in the first row</td>
    <td>second cell in the first row</td></tr>

<tr><td>first cell in the second row</td>
    <td>second cell in the second row</td></tr>

<tr><td>first cell in the third row</td>
    <td>second cell in the third row</td></tr>
</table>
```
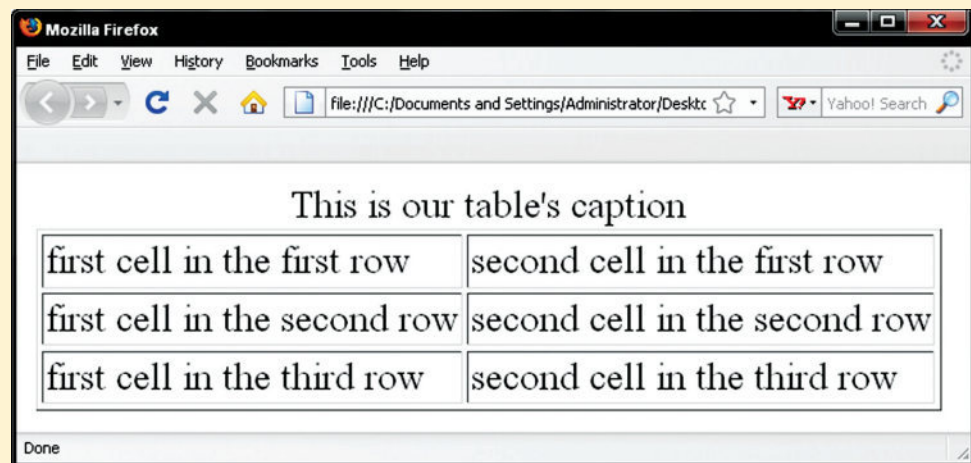

Figure 9.11 : A table with 3 rows and 2 columns

The basic table creation uses the following 5 tags:

\<table> \</table>	This pair defines the beginning and end of a table
\<tr>\</tr>	This pair (\<tr> stands for table row) defines a row of the table. So each row of your table requires one pair.
\<td>\</td>	This pair (\<td> stands for table data) defines a cell and number of such pairs you use will define the number of cells your row is divided into.
\<th>\</th>	Used in place of the \<td>\</td> tag, this pair (\<th> stands for table row Heading) formats the text in a cell as bold face and centre it to give the effect of a row heading.
\<caption> \</caption>	This tag pair is used to provide a heading to the table.

These basic table tags can be expanded to produce a variety of tables. Other HTML tags for highlighting and formatting text can be used along with their attributes and values. Let us examine some of these attributes, their possible values and the effect it has on the table.

Attributes of the \<Table> tag

Specifying the table border and border colour

The **BORDER** attribute of the \<TABLE> tag is used to give the table a border. The BORDER attribute specifies the thickness of the table border as a number. The default setting is BORDER=0, which means, if you do not use the border attribute, you will end up with a table with no border.

Example : **\<table border=2>**This table has a border of thickness 2**\</table>**

There are two ways of colouring a table. You could colour the border or you could fill the cells with colour. You use the **BORDERCOLOR=** attribute to specifies a colour for the table border. You use the **BGCOLOR=** attribute to specify a colour to fill.

Example : \<table bordercolor="red">This table will have a red border\</table>

\<table bgcolor="green">The cells of the table will have a green background\</table>

Like we saw earlier, the value of these attributes could also be the hexcode of the colour. So you could use #0000ff in place of blue. This is very useful when you want to use a particular shade of a colour and that does not have a name.

Instead of a colour, you could also use an image as a background in the table. The **BACKGROUND** attribute specifies the image to fill the background of the table or of individual cells. While specifying these attributes, remember :

247

- in the <table> tag, it affects the entire table (cells as well as the margins).
- in the <tr> tag, it affects the entire row (cells only). It overrides the colour specified in the <table> tag.
- The **BACKGROUND** attribute does not work if specified in the <tr> tag.
- in the <td> tag, it affects the specified cell. It overrides the colour/image specified in the <table> or the <tr> tag.

Adjusting the sizes of the table and its cells

When no sizes are specified the table is only as big as it needs to be according to its contents. The attributes **WIDTH** and **HEIGHT** can be used to control the width and the height of the table respectively. These attributes can be specified either as a percentage (of the browser window) or as a specific size measured in pixels.

There are a few peculiarities related to these attributes, when you use them with the <tr> and <td> tags that you should bear in mind.

- Using the HEIGHT attribute in a particular <TD> or <TR> tag overrides the HEIGHT specifications of the <TABLE> tag.
- Using the HEIGHT/WIDTH attribute in a particular <TD> or <TH> tag causes a height/width change in all the <TD> tags of that column. In case two cells of the same column contain HEIGHT/WIDTH attributes, then normally the browser uses the higher width specified.

Customising Margins

The attribute **CELLPADDING** specifies (in pixels) how much empty "space" exists between items in the cells and the walls of the cells, a higher value for this attribute making the cells larger, "padding" the cell.

The attribute **CELLSPACING** specifies (in pixels) the distance between two adjacent cells. This is particularly useful when you use the table tag to separate blocks of text. For instance you could use this attribute to create a two column text.

You will notice that if you do not specify a cell padding, the table border will stick to the edge of the contents of the table. Some examples are given below in figure 9.12.

| <table cellpadding=2> | <table cellspacing=2> | <table> |

Figure 9.12 : The use of cell padding and cell spacing attributes

Customising Cell Alignments

Cell alignment refers to the horizontal or vertical alignment of text in a cell. By default the cell contents are left aligned (horizontally) and centered (vertically). To change this, we use the align = attribute in the <td> tag. Look at these examples:

For horizontal alignment,

<TD ALIGN=left> aligns text to the left

<TD ALIGN=right> aligns text to the right

<TD ALIGN=center> aligns text to center

For vertical alignment

<TD VALIGN=top> aligns text to the top

<TD VALIGN=bottom> aligns text to the bottom

<TD VALIGN=middle> aligns text to the middle

You can also combine a horizontal align = attribute with a vertical align = attribute.

Merging Rows and Columns

Merging of two or more adjacent cells or two or more rows can be achieved using the **ROWSPAN/COLSPAN** attribute. This helps us stretch a cell across the entire row, or over multiple rows or columns. Look at the examples below (Figure 9.13):

| <tr rowspan=2> </tr> | <tr colspan=2> </tr> |

Figure 9.13 : The use of rowspan and colspan attributes

Nesting tables

Try placing two tables side by side on a web page. You will notice that the <table> tag automatically sends the second table to the next line. To get around this problem, we simply nest the two tables inside a third table, a sort of border around the two tables, thus keeping them on the same line adjacent to each other.

249

You create a table with one row and two columns and put in a table into each of the two cells. See the code and the result in figure 9.14 below.

```
<html>
<table border="1">
<tr>
<td>
<table >
<tr>Table1</tr>
</table>
</td>
<td>
<table ><tr>Table2</tr></table>
</td>
</tr>
</table>
</html>
```
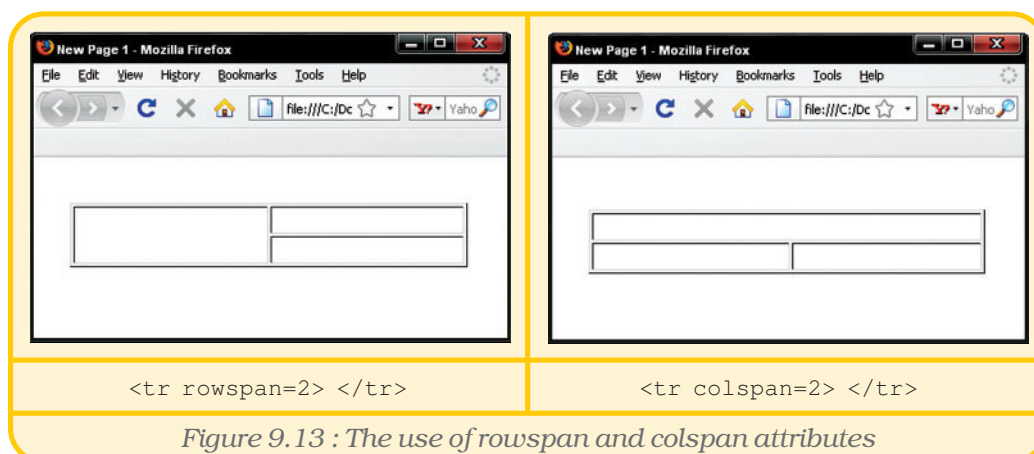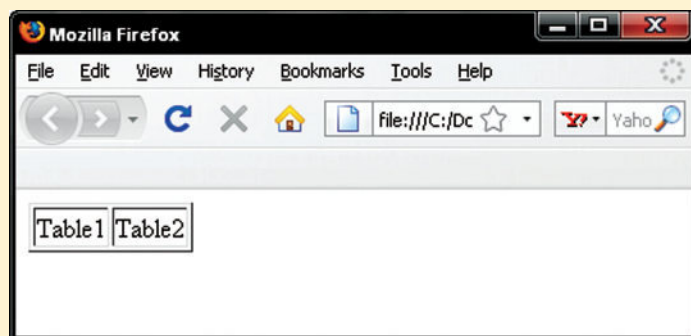
Figure 9.14 : Code of table within a table

Nesting tables can also be used to create tables where each table row has different cellpadding and cellspacing. Try this out.

Let us now examine the use of tables in our web page. Look at figure 9.15. There are two distinct ways in which tables are used here. The overall web page itself has a table like look. This table contains two rows. While the first row has only one cell or column, the second, containing the menu options has five cells. The third which constitutes the text of the web page has two cells or columns.

Figure 9.15 : Use of tables for organising content on a web page

You also have three tables visible on the page :

- The first table listing the contents has six rows or a single cell each. This table is nested within the first column of the third row of the main table.
- The second listing the specifications and prices of hardware, is nested in the second column. While the header row indicates the types of information, the first column lists the type of hardware.
- The third table in the third column of the third row of the main table holds thumbnails of an image gallery. Each image is a hyperlink to a larger version of the image and will open on another web page. We will discuss the tags for this table in the section 9.5.10.

If you observe the source code of this page, you should be able to identify the tags for these tables. Write down the tags for these tables. Take care to ensure that each tag has its corresponding closing tag. This is particularly necessary while nesting tables, rows or cells. Also notice the use of various attributes and their values. Vary them and observe the effect.

9.5.10 PHOTOGRAPHS AND OTHER MEDIA ON THE WEB PAGE

Most of the websites you encounter would have one or more images. It could be the use of a logo of the organisation, some design elements, a background, an image gallery or even a photograph used for beautifying the web page.

Images can be drawings, photographs, graphs, or even decorated text saved as images. Keeping in view the need for small file sizes, HTML supports jpg, gif, and png formats. But as we learnt earlier, HTML produces a text file. Hence images are not placed inside the file. A reference to the file is made and the image file accompanies the text file.

We use the <img> tag to insert an image on the web page. This is always used with the attribute SRC, which stands for the source of the file (what is the image file and where is it located?). Typical examples are :

```
<IMG SRC="image.jpg">
<IMG SRC="\images\image2.gif">
```

In the first example, the image referred to lies in the same folder as the html file. In the second example, the image is in a sub-folder called images, which is in the same folder as the html file. This also brings about an interesting option. The image file can be anywhere, as long as we can define its exact location. The image can even be on a different web site, residing on a server half way across the world. The tag would look like :

```
<IMG SRC="www.mywebsite.com/photographs/photo1.jpg">
```

Of course this assumes that the website and the image is available on the remote website, when you are viewing your web page.

Attributes of the <IMG> tag

The <IMG> tag can also take many attributes, which allow formatting, highlighting and specifying the size of the image.

Specifying Image Dimensions

Remember that images can have a large file size. Specifying an image's dimensions (height and width) can control the loading time of images. As the browser loads the page, it notes the height and width attributes and leaves that much space for the image. Next, it lays out the remaining

text, and then it goes back and fills in the image. If these attributes are not specified, the browser has to load enough of the image to get the dimensions before it can lay out the rest of the text, thereby slowing the display of other page elements. To specify image height and width, add the **HEIGHT** and **WIDTH** (in pixels) attributes in the <IMG> tag.

What happens if the height or width values are smaller or greater than the image? Simple, the image gets cropped (cut) or stretched (Figure 9.16). So, height or width should exactly match the actual image.

Width specified is less than image width	Width specified is more than image width	Width specified matches image width

Figure 9.16 : Effect of specifying image dimensions
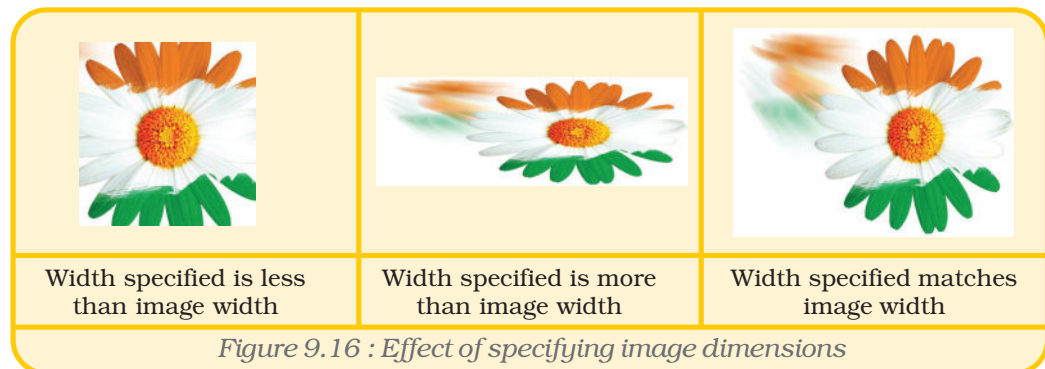
Aligning Images

By default the image is aligned to the left of the web page and the text is aligned at the bottom most line of the image. To change this default setting, HTML offers several image alignment options:

• Three options vertically align the image with respect to a line of text: align=top, align=bottom and align=middle
• Two options align the image to the left or the right of the window (with corresponding text wrap)

Look at figure 9.17 below.
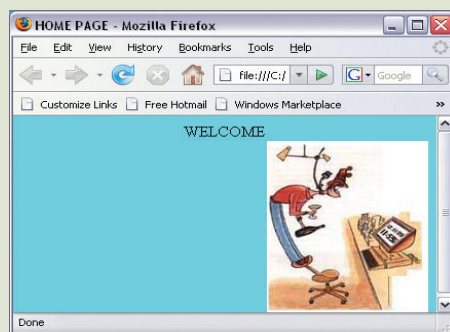
The ALIGN=TOP aligns the top of the image with the top of the surrounding text as displayed in the adjacent figure.

ALIGN = MIDDLE places the middle of the image at the baseline of surrounding text.	
Similarly, ALIGN = BOTTOM places the bottom of the image on the same line as the text. This is the default setting.	
ALIGN = LEFT places the image on the left of the browser window.	
ALIGN = RIGHT places the image on the right of the browser window.	
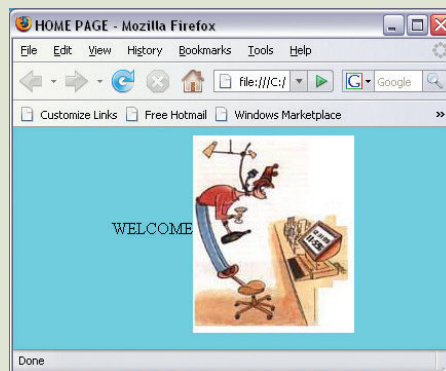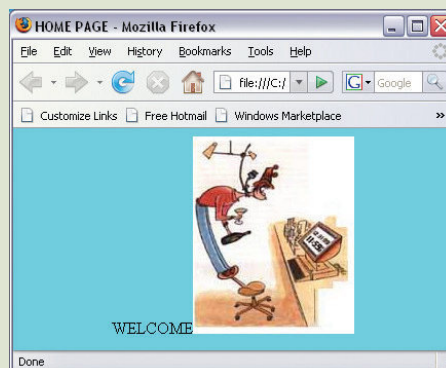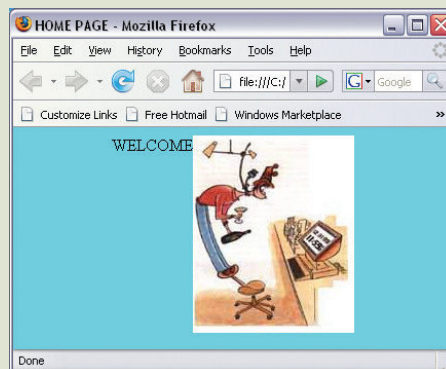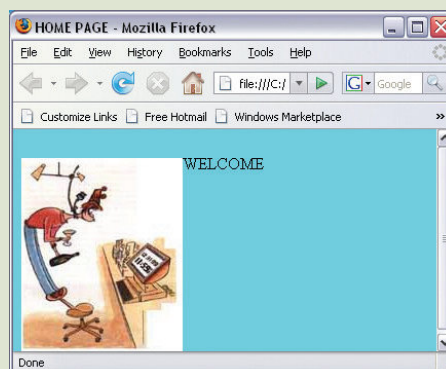
Figure 9.17 : Use of align = attribute for the image tag

253

Image Borders

The **BORDER** attribute of the <IMG> tag displays a border. You can also control its thickness by specifying a value. The default value is 0, indicating no border. If you do not specify a border colour using the bordercolor= attribute, then the colour specified in the <body> tag will be used. If you have not done so, then the default colour, black will be used.

Displaying Alternative Text

The web is used by all people and a few of them could be blind. How do blind people use the web? Screen reader software read the page for them. But what happens when you encounter an image?

To facilitate this, the <IMG> tag uses the **ALT** attribute. It is used to specify the **Alt**ernative text to be displayed. This attribute can also be useful if :

- The user uses a text-only browser.
- The user turns off images so that a file loads faster.
- The browser does not display images correctly.
- Images are not displayed because links aren't working properly.
- Text is to be displayed while images load.

Now let us look at our web page and look at how the image gallery was created. Look at the source code for this table in figure 9.18. Nothing extraordinary. The table has five rows with five cells each, all of them equally spaced.

```
<table border="1" width="735">
 <tr>
   <td width="245" height="245"><img src="1.jpg" height="245" width="245" /></td>
   <td width="245" height="245"><img src="2.jpg" height="245" width="245" /></td>
   <td width="245" height="245"><img src="3.jpg" height="245" width="245" /></td>
   <td width="245" height="245"><img src="7.jpg" height="245" width="245" /></td>
 </tr>
 <tr>
   <td width="245" height="245"><img src="4.jpg" height="245" width ="245" /></td>
   <td width="245" height="245"><img src="5.jpg" height="245" width="245" /></td>
   <td width="245" height="245"><img src="6.jpg" height="245" width="245" /></td>
   <td width="245" height="245"><img src="8.jpg" height="245" width="245" /></td>
 </tr>
</table>
```

Figure 9.18 : Source code for the image gallery

As expected, each <tr></tr> pair has five <td></td> pairs within in, representing the five cells in each row. Each cell contains an image and the image itself functions as a link to another image. Therefore you should expect a code of the form <img src=img01.jpg>, where the <a> pair refers to the hyperlink and the <img> tag refers to the thumbnail. Observe the source code and confirm it.

9.5.11 MAKING THE WEB PAGE INTERACTIVE – HYPERLINKS

What makes the Internet one of mankind's greatest resources is its ability to link resources to one another and on the click of a mouse, you can jump to a page residing on a machine half way across the world. We call these **hyperlinks.** In this section, we will explore the tags that makes this happen. This would be one of the simplest, but still one of the most significant tag pair <A>text</A>.

The text between the tags is converted to a link. This link could be a link to another paragraph on the same page, another page in the same folder, another page in some other folder or even a page in some other machine, elsewhere. But on its own this tag would be useless, as it still doesn't say, which page it links to.

The attribute that enables this addressing is HREF=. So, the hyperlink tag looks like :

```
<A HREF="http:/mywebsite.com/mywebpage.htm">link to my site
</A>
```

What does this say? The HREF= attribute has a value which refers to a page "mywebpage.htm" on the site "mywebsite.com". As long as this page actually exists on this site and the server hosting this site is connected to the web, you will be able to call this page – a click on the link would display this page.

The World Wide Web uses an addressing scheme known as **URL** to indicate the location of files on the web. URL stands for **U**niform **R**esource **L**ocator (sometimes also called "Universal Resource Locator"), which is the address of a document on the Internet. You will notice the URL for your web page displayed in the address bar of your Web browser.

Hover your mouse cursor over a hyperlink. Notice that the cursor changes shape into a pointed finger icon. At the same time, you will see the address to which this hyperlink points in the status bar at the bottom-left corner of the page.

Types of URLS

URLs are of two types, absolute URL and relative URL.

An absolute URL contains the complete address of a file on the Internet. An absolute URL therefore refers to the hostname, folder

255

name and filename, and also the protocol (set of rules) it will use to connect to the web page or file.

For example: http://www.ncert.nic.in/html/schoolcurriculum.htm

Notice, we keep referring to the file on the Internet. The website not only hosts web pages, but also images, video, audio, documents, zipped files or folders, and practically any type of digital file. Browsers mostly interact with web pages, which are files of the html type. All other files are downloaded on to your machine (or executed if it is a program).

A relative URL, on the other hand, is a shorter form, and points to files on the same server or in the same folder. A relative URL contains only the folder name and filename or just the filename if the web page containing the link and the resource called are in the same folder.

For example: /html/school.htm

This URL refers to a file school.htm residing in a folder html. The folder resides on the same site as the web page containing the link. We could have written the complete address as

http://www.mywebsite.com/html/school.htm but the shorter form is sufficient.

Types of hyperlinking

As we mentioned earlier, hyperlinks can be to a destination within the same page, or to some other page. We could also use hyperlinks to launch scripts or applications and perform other functions, for example, we could launch a slide show, play a movie or even send an e-mail. Let us examine the attributes for these actions.

External linking

This is perhaps the most common usage of hyper links. Page1 containing the link hyperlinks to Page 2. And as we mentioned earlier, page 2 can be on the same site or on a remote site.

The format is <A HREF="http://mywebsite.com/myfile.htm">link to my file</A>. If the destination file, in this case myfile.htm, happens to be a web page, then the browser handles it on its own. The new web page is displayed.

If the destination file is let us say, a video, then the browser calls upon a media player to help display the contents. We call such add on software as plugins. Java scripts are also used in some cases. Browsers can display text and images, but for all other media they would require external support. So, you could have plugins to play video, audio, animations, or scripts to play slide shows or even an executable (.exe) program.

Let us consider an example of playing a video clip, using the Windows Media Player. You will have to embed the following code in to the <body></body> tag at the top of the web page.

```
<embed type="application/x-mplayer2" pluginspage="http://
www.microsoft.com/Windows/MediaPlayer/" name="mediaplayer1"
autostart="false" width="320" height="240" src="myvideo.wmv" />
```

Be sure to have a video file named myvideo.wmv in the same directory as the html file, or simply change the src attribute to point to the path of the video file.

Let us review the attributes of the <embed> tag :

• type="" defines the type of embedded object you'd like to use. This tag is mandatory for Windows Media Player to be embedded in the web page;
• pluginspage="" this attribute is needed only if the user computer does not have a Windows Media Player plugin; as you cannot be sure, include it;
• name="" this attribute assigns a unique name to this embedded object, which will allow you to place and access multiple embedded objects (see the section on named anchors below);
• autostart="" defines whether or not the video should start playing automatically when the page is loaded;
• width="" and height="" defines the size in pixels of the Windows Media Player embedded window. If you want the visitors to see the video at its original resolution, omit the width and height attributes.
• src refers to the source, it defines the path to the file you want to play.

Internal linking

As we mentioned earlier, the link could be to another part of the same page. Typically, you would need this when you have a contents page listing different sections of a document and the sections themselves, all in the same web page.

You would require to mark particular sections of a page using a reference marker. This is done using the NAME attribute. Let us look at this example :

Code	Output
`<html>`	
`<A NAME="CONTENTS">Contents </A>`	Contents
`<A HREF="#chap1">Chapter 1</A>`	Chapter 1
`<A HREF="#chap1">Chapter 2</A>Chapter 2`	Chapter 2
`<p>Introduction:`	Introduction:
`Text of Introduction`	Text of Introduction

`<A NAME ="chap1">Chapter 1</A>`	Chapter 1
`Text of chapter 1`	Text of Chapter 1
`<A HREF="# CONTENTS">Back to contents</A>`	Back to contents
`<A NAME ="chap2">Chapter 2</A>`	Chapter 2
`Text of chapter 2`	Text of Chapter 2
`<A HREF="# CONTENTS">Back to contents</A>`	Back to contents
`</html>`	

From the output above, you will notice that the tag pair and the attribute <A NAME=" "> </A> and the tag pair and attribute <A HREF=" "> </A> are converted to links. Let us examine what these links refer to.

A click on the links Chapter 1 or Chapter 2 leads to the beginning of the respective chapters. The link at the bottom of each chapter Back to contents will lead back to the contents list. The tag pair <A Name=" "></A> is called a named anchor and is actually not visible on the web page. This tag functions as the destination for the link. So, when you click on the link Chapter 1, you are led to this anchor. Similarly, when you click on the link Back to contents, you are led to the anchor for the contents list.

The other tag pair, <A HREF=" "></A> is similar to any other link, but the destination reference is preceded by a hash (#). This hash prompts the browser to look for a named anchor on the same page.

This technique of navigation within the page is very useful when you wish to keep a document together and not split it into many different web pages. Printing, or downloading this page will yield the entire document in one piece. While it may look silly on short web pages, it is certainly very useful if you have a very long web page and have to frequently go back and forth or seek particular sections.

You can also use this technique when you wish to refer to a particular section on a different web page. In this case, the hyperlink will refer to a different web page and a named anchor on that web page. See the example below :

```
<a href="index.html#section2">
```

The link refers to an anchor named section2 on a web page named index.html.

Using images as links :

Large images have big file sizes and take time to load. To overcome this, we represent the image with a smaller image on the web page. We refer to this as the thumbnail. So we would like the user to click on this image to see the larger image. Thumbnails can be used to represent

not just images, but any resource. For instance, you may like a photograph of the cover page of a book to represent the web page containing the book.

The tags used in this case is a combination of the <A HREF=""></A> and the <IMG SRC=> tags. Look at this example:

```
<A HREF="mywebpage.htm"><IMG SRC="myphoto.jpg"></A>
```

You would have used the tag <IMG SRC="myphoto.jpg"> to display the picture. Enclosing this tag with the <A HREF=></A> tag, converts this picture itself into a link. So a click on this link would lead you to a new page, mywebpage.htm.

Notice, that it does not say anything about what the new page contains. It simply links to a different page. In fact, this facet of HTML allows it to hand over control to the next page, leaving that page to handle resources relevant to that page.

E-mail links

If you wish someone to comment about your web page or the content on it, or you wish to invite someone to contact you, an easy way of doing so would be to put in your contact details (address or phone numbers). An easier way would be to provide an e-mail link so that an e-mail can be sent to you from right there. The tag to use is still the <A HREF=></A>. See this example:

```
<A HREF="mailto: dceta.ncert@nic.in">Contact Us</A>
```

The "mailto:" part of the attribute launches your default mail client (see figure 9.19 below). Of course, you should have configured your mail client to send mails. If you usually use a browser based mail programme like Hotmail, or Yahoo, or Gmail, you may not have configured it. Typically, in a Windows environment, Outlook Express is the default e-mail client, unless you have configured some other mail client. So, this tag will be useful and serve the purpose only on those machines, where an e-mail client is configured. Nevertheless, it is a very useful and time saving tag attribute.
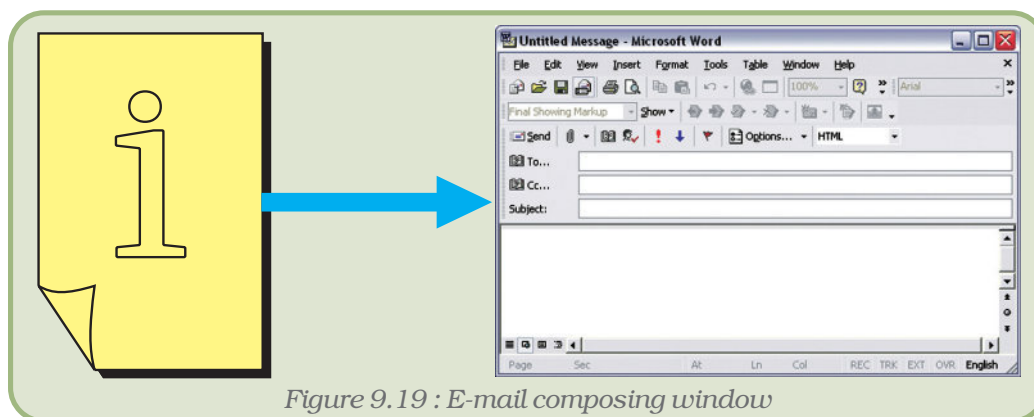

Figure 9.19 : E-mail composing window

259

Specifying an e-mail address within the anchor tags instructs the browser to automatically insert the mentioned address in the "to" text box of the e-mail composing window.

9.5.12 Making The Web Page Interactive – Input Forms

In the previous section, we explored web interactivity using the <A></A> tag pair. Hyperlinks are only one form of interactivity that the web provides. It does allow us to jump around from one page to another, but all this is predefined. What if you wish to seek an answer to a question!

This would require a way of sending your question to the server and some way in which the server processes your question and respond to it. HTML provides for such forms of interactivity too. There is a range of types of inputs that can be submitted. Not only does it allow a web site to respond to user queries but also seek information from the users.

The tag pair used is <form></form>. Let us explore its use. Primarily this tag must permit the input of text. It must then define a delivery mechanism and the destination for this message. This is achieved by the attributes ACTION and METHOD respectively.

The ACTION attribute specifies where the information will be sent for processing. Information processing destinations can be CGI programs, JavaScript functions, or an e-mail address. **CGI** or the Common Gateway Interface (CGI) is a standard protocol for interfacing external software with an information server, commonly a web server. Look at these examples :

Action	*Code*
Sending request to a cgi program	ACTION="/cgi-bin/something.cgi" (relative url) or ACTION="http://mywebsite.com/cgi-bin/something.cgi"(absolute url)
Sending request to a javascript	ACTION="javascript:SomeFunction()"
Sending request to an email address	ACTION="mailto:me@mywebsite.org"

There are two methods of sending information in a form to the server, the get and the post method. We use the METHOD attribute of the <FORM> tag to specify this. Of course, the server and the associated software will predetermine what method you will use and so you may not have a choice. But it is useful to know the difference between the two methods.

With the "get" method, the query or information along with relevant codes is appended to the URL specified by the action attribute and this

new URL is sent to the processing agent. The "get" method should be used when the form is idempotent (i.e., causes no side-effects). Many database searches have no visible side-effects and make ideal applications for the "get" method. Further the get method restricts the input to ASCII characters.

So, if you searched for solar eclipse 2009 on mywebsite.com, the form submitted would read http://www.mywebsite.com/search?q=solar+eclipse+2009.

With the "post" method, the query or information is included in the body of the form and sent to the processing agent. If the service associated with the processing of a form causes side effects (for example, if the form modifies a database or subscription to a service), the "post" method should be used.

Form Elements
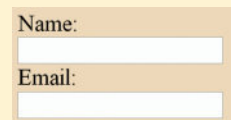
Having defined the action and the method, we define the format of the information to be input and sent to the server. These can be in the form of text entry fields, check boxes, radio selections, password entry fields, form buttons, file upload fields, or image buttons. All these are defined as attributes of the **<INPUT> tag.**

The <INPUT> tag can have several other attributes depending on the value of the "TYPE" attribute. The type attribute specifies the type of input we want from the user. It can be used to create the following type of interface elements:

1. Text field
2. Password field
3. Radio button
4. Check box
5. Submit button
6. Reset button
7. Image button
8. Hidden field

The **TYPE="TEXT"** attribute is a common form input field. It creates a place wherein one line of information can be typed.

```
<LABEL for="firstname">Name: </LABEL><br>
<INPUT type="text" id="firstname" size="20"><BR>
<LABEL for="email">Email: </LABEL><br>
<input name="text" type="text" id="email" size="20">
```

Name:

Email:

The **TYPE="PASSWORD"** attribute works the same as the type="TEXT" attribute except that the characters typed into the input area are displayed as asterisks (or some special character). Password text is scrambled during transmission and then unscrambled when the form data is received at the server end.

Radio buttons are used when the user is expected to select one – and just one – option from a set of alternatives. The **TYPE="RADIO"** specifies a set of controls that are linked so that only one radio button among each set is selected at a time; if the user clicks one radio button, the others in the set are automatically de-selected. All options for the radio button should have the same name. The value sent by the form is the value of the radio button that was last selected. Adding the option **CHECKED** to one of the buttons in a set will make that button pre-selected when the page loads, a kind of default value. Look at the example below:

```
<LABEL for="firstname">Name:
</LABEL><br>
<INPUT type="text" id="firstname" size="20"><BR> <LABEL
for="email">Email:
</LABEL><br>
<input name="text" type="text" id="email" size="20">
Gender<BR>
<INPUT type="radio" name="sex" value="Male">
Male <INPUT type="radio" name="sex" value="Female"> Female
```

Name:

Email:

Gender
  ○ Male ○ Female

Check boxes are used to allow the user to select multiple options from a set of alternatives. A check box is thus not affected by other buttons, so it is possible to have more than one in a group checked at a time. They can be created using the **TYPE="CHECKBOX"** attribute. Notice that unlike a radio button, every check box has a unique name. Clicking it will place a check mark ("X" or "”") and clicking it again will remove the mark. The value sent in the web form is the value of the check box if it was selected; otherwise the value will be empty. You may add the option **CHECKED** to a particular check box to define it as the default value.

```
Language<br><input type="checkbox" name="option1"
value="hindi">Hindi<input type="checkbox"
name="option2" value="english" checked>English
```

Language
  ○ Hindi ● English

The **TYPE="SUBMIT"** creates a submit button on the form and the **TYPE="RESET"** creates a reset button. The Submit button tells the web browser to gather up all the selections, values, and entered text in the form elements and dispatch it off to the destination defined in the **ACTION** part of the <FORM> tag. The Reset button restores the form to its default state.

```
<INPUT type="submit" value="Send"><INPUT type="reset">
```

Send   Reset

The submit or the reset button has a typical shape and size, which may not be in harmony with the rest of the look and feel of the web page. Hence you may prefer to use an image instead using the **TYPE="image"** attribute. When a user clicks an image button the

form is sent to the address specified in the action attribute of the <form> tag.

```
<p align="center">
  <INPUT TYPE="image" SRC="flower.jpg"
NAME="imageclick"
        ALIGN=top HEIGHT=50 WIDTH=50>
This image works like the submit button
</p>
```

This image works like the submit button

Alternately, the **TYPE= "BUTTON"** creates a click-able button with text the user specifies in the "value" attribute. When the button is clicked, it fires the **On Click** event. This event can be processed by, any scripting language such as VBScript or JavaScript to do some specific task. Of course if the event is not programmed nothing will happen on clicking the button. The **Submit** and **Reset** buttons, for example are special types of buttons that have been pre-programmed to do specific tasks.

```
<INPUT TYPE="Button" VALUE="I am a simple button">
```

I am a simple button

The **File** field allows users to upload files. The user can submit any information like picture, a spreadsheet, a scanned document or a word-processor document after verifying that the server on which the form will be processed supports file upload. The browser displays a "BROWSE" button next to a text box that lets the user select a field from their computer's storage to use as the value of the file input element. Typically, you would have encountered this option when attaching files to your email messages.

The output produced will be:

```
<INPUT TYPE ="FILE" NAME ="name"
SIZE ="30" MAXLENGTH ="200">
```

Browse...

A hidden field is similar to text field with one major difference. The hidden field does not show on the page. Therefore the user cannot type anything in the hidden field. The **TYPE="hidden"** attribute is used to specify information for the receiving program or function that is not visible to the user and that the user may not change. This information is usually required by the program or function in order to process other information.

```
<INPUT TYPE="hidden" NAME="Language" VALUE="English">
```

The hidden field does not show, but still, when the form is submitted the hidden field is sent with it. In the above example the hidden field would tell the program that handles the form, that the users preferred language is English. Bilingual or multilingual sites, for example, can then switch to the correct language.

263

The <SELECT> tag

The **<SELECT>** tag is used to construct drop-down list boxes (also called drop-down menus) and scrolling list boxes (also called scrolling menus). Drop-down menus can serve the same purpose as radio buttons (one selection only) or check boxes (multiple selections allowed), depending on the options specified. The advantage of a drop-down menu, compared to radio buttons or check boxes, is that it takes up less space. But that is also a disadvantage, because people can't see all options in the menu right away. With the **SIZE** attribute, the menu can be customised to show more than just one option at a time. Such lists are typically encountered when the user is to select a city or a country.

The **<OPTION>...</OPTION>** tag is used to specify each item displayed in the list. The value of the option last selected is returned when the form data is transmitted. Adding the **SELECTED** attribute in the <OPTION> tag indicates which element is displayed initially when the page loads. If this is not provided, the first item is selected by default.

```
<select name="abc"><option value="Math">Math
</option><option value="Science">Science
</option></select>
```

Math :

The above listing will display a drop down list since the size attribute is not specified:

The <TEXTAREA> tag

Text areas are text fields that can span several lines. They are used for accepting multi-line input from the form user. Unlike most other form fields, text areas are not defined with an <INPUT> tag. Instead we enter a <TEXTAREA> tag where we want the text area to start and a closing </TEXTAREA> tag where we want the area to end. Everything written between these tags will be presented in the text area box. The scrollbars will be activated automatically, if the text input exceeds the visual number of columns or rows specified.

The above listing will display a text area with 2 rows and 20 columns as displayed in the following output:

```
Address<br><textarea name="as" rows="2"
cols="20"></textarea>
```

Address

9.6 EXTENDING INTERACTIVITY – TRENDS IN WEB TECHNOLOGIES

We learnt about two kinds of interactivity in this chapter, hyperlinks and input forms. Web technologies have developed rapidly and today support a range of other features, unheard of a few years ago.

Web sites today support interactive forums, rich multimedia applications, and even radio stations. Web sites support commercial transactions like banking, bookings and reservations, and shopping. All such web features require large amounts of information to be stored and retrieved. They should also support different information for different users simultaneously.

The Indian Railways reservation system is a good example. Different people would request tickets for different places, by different trains, by different classes, on different dates. Simultaneously the system should make available information to each of these users, storing their names, ages, etc. and charge them accordingly. This charging is also done on the web by interlinking with banks. It is impossible to create different web pages for each possibility. What is required is a technology by which, based on the user's needs a new page is created by the server and sent to the user's browser. We refer to these as dynamic web pages.

9.6.1 DYNAMIC WEBSITES

Web sites containing dynamic web pages are supported by databases. The web page itself contains empty tables which are populated based on the users request. Let us consider another common example, a search engine, say Google. You put in a search word or phrase and click on the search button. Within seconds, the web page is reconstructed with lists of web sites containing information you sought.

We refer to the search word you input as the keyword. Let us say, your keywords were "next solar eclipse in India". What you actually sought to know was "When will the next solar eclipse occur in India?". So, when you clicked on the search button, you actually submitted a request to the server to answer this question. We refer to your request as a query. You queried the server. The server looked up the databases for information matching your query and reported back with information relevant to your query.

Not only search engines, but a variety of different web applications use dynamic sites. Online libraries, dictionaries, encyclopedias, and book stores require dynamic features. E-mail providers, social networking sites, online courses, shopping malls, and tourist bureaus also require dynamic features. In fact, any site which seeks to provide constantly changing information, and different information for different users require the support of dynamic technologies. Programming technologies like active server pages (asp), java server pages (jsp) and hypertext preprocessor (php) are used to construct dynamic web sites. Such sites require a database to hold the information and a server to query, retrieve and serve the results of the query to the user.

9.6.2 SEPARATING CONTENT, STYLE AND CODE

When you look at the html coding of our web page, you would have noticed that the coding, the design elements – like the tables, fonts

and colours, and the content of the web page are all put together. Not only did this make web pages slow to load, but also made it difficult to change. It required the graphic designer, who was mostly concerned with the look and feel of the web page, the subject expert who was interested in the content of the web page and the programmer, who handled the html code to work closely together. Each time a new page had to be made, all the three had to get together again.

Developing dynamic sites with thousands of different pages would have been very difficult to make and manage using such methods. An interesting technology has allowed us to separate the design, the content and the programming elements.

This technology, known as cascading style sheets allows the placing of all the design elements – colours, images, positioning, fonts, sizes and layouts to be placed in a separate file, the style sheet file. The content is placed in a text file or as records of a database. The html code along with other advanced code for supporting dynamic features is placed in the web page. The style sheet and the web page together function as a template, a container with predefined style elements. When the user calls for a page, the content, the style and the code are combined dynamically to generate a web page.

The use of cascading style sheets and databases now allows you to concentrate on content development. Once developed, the style sheets, the database and the web page code will remain fixed, unless you wish to modify it. New content gets populated in to the database, making it very easy for web site content to be renewed frequently.

Summary

- One of the major applications of the Internet is the World Wide Web (www).

- The web can be thought of as an interconnected set of documents, images, audio, video or software files.

- The web uses the Hyper Text Markup Language to request and serve web pages and programs. Browsers are designed to read and interpret HTML.

- HTML consists of tags; the general format for a HTML tag is <tag_name>affected text or image</tag_name>

- Tags have different attributes and attributes take different values. Together, they extend the scope and functionality of HTML tags.

- You will need a HTML editor to write your code and a browser to test your web page; browsers differ in the way they interpret specific tags; it is therefore advisable to test your web page on different popular browsers.

- The basic structure of a web page is
```
<html>
    <head> </head>
    <body> </body>
</html>
```

- The code that goes into the <head></head> tag pair is not displayed by the browser; the title tag is an exception; it is displayed in the title bar of the browser window.

- All references to scripts, fonts, meta data are made in the head section.

- All elements of the web page visible in the browser window go into the <body></body> tag.

- Tags associated with size, colour, font are used to highlight text elements on a web page; You can also highlight text using bold, italic or underline.

- Text can be organised into paragraphs and paragraphs or lines of text can be separated from each other with the use of <p></p> and <br> tags.

- Lists can be prepared and presented using the ordered or unordered list tags; list items can be numbered in different ways using the type attribute.

- The <table></table> tags along with its associated tags and attributes is used in two ways; to separate different blocks of content and to present content in the form of tables.

- Tables consist of rows and cells; they can be structured and beautified using backgrounds, images, controlling their sizes and contents; tables can be nested within other tables.

- Images, audio and video files can be embedded in web pages. While browsers display images on their own, they require the support of media plugins to play back audio and video files.

- Hyperlinks are the most powerful feature of the web. Hyperlinks allow us to link two resources. The resources can be on the same or a different page, on the same storage or on a computer located half way across the world.

- Hyperlinks are achieved using the link tags.

- Forms can be used to submit information to, place a request on and retrieve information from the server. This extends the interactivity of web pages.

- Linking web pages to databases have enabled content on request, whereby servers respond to specific queries of users and dynamically generate web pages containing responses to the queries.

EXERCISES

SHORT ANSWER TYPE QUESTIONS

1. Identify and name the following:
   (a) One physical tag and an equivalent logical tag used for making the text bold.
   (b) One physical tag and an equivalent logical tag used for making the text appear in italics.
   (c) A Windows accessory used as a text editor for creating HTML documents.
   (d) Two possible extensions of an HTML file.
   (e) A tag used for inserting section breaks.
   (f) A tag used for displaying headings in a web page.
   (g) Different tags necessary for creating tables.
   (h) Tags used for creating framesets.
   (i) Mandatory attribute of the <A> tag.
   (j) An attribute of the <A> tag used to specify name of the frame in which the linked file is to open.
   (k) Two attributes of the <FRAME> tag used to control the amount of white space between the frame borders and the contents of the frame.

267

2. Define the following terms:
   (a) WWW
   (b) Hypertext
   (c) Markup
   (d) Definition List

3. Differentiate between:
   (a) Container and Non-container tags
   (b) Text editors and WYSIWYG editors
   (c) Logical and Physical tags
   (d)
 and <P> tags
   (e) Ordered List and Unordered List
   (f) Internal and External Linking
   (g) <A> as an anchor and <A> as a link
   (h) Cellpadding and Cellspacing attribute of the Table tag
   (i) Align and Valign attribute of <TD> tag
   (j) <TD> and <TH> tag
   (k) <TD> and <TR> tag
   (l) <FRAMESET> and <FRAME> tags
   (m) FRAMEBORDER and BORDER attribute of the <FRAMESET> tag

4. Correct the errors in the following HTML codes:
   (a)
   ```
   <HTML>
   <T> Term Test </T>
   <BODY>
   <FONT SIZE=6>My Shopping List</FONT>
   <BREAK>
   <OL Number="A" START=D>
   <LI>Shoes
   <LI>Socks
   </OL>
   </BODY>
   </HTML>
   ```

   (b)
   ```
   <HTML>
   <TITLE><HEAD>Using Link</TITLE></HEAD>
   <BODY>
   <A Name="#top">Links</A>
   <P>This page is an example of using links
   using the anchor tag. </P>
   <A Href="top">Goto top</A>
   </BODY>
   </HTML>
   ```

5. Identify the tag and attribute(s) used:
   (a) To mark up text to retain its formatting along with blank spaces, carriage returns etc.
   (b) For creating a list which uses alphabets as the numbering style and starts with the alphabet D.
   (c) To insert an image as a background of a web page.
   (d) To insert a line along the width of the web page to break up long sections of text that is 10 pixels thick.
   (e) For linking within a web page.
   (f) Create named frames for linking.
   (g) Creating a link to a specific point named MARKS in a web page named Result.html stored in the same directory as your web page.
   (h) For displaying "Welcome" as a tool tip text for an image with a 7 pixels thick border.

   (i) For creating an e-mail link to send a message to ncert@gmail.com.
   (j) To display the cell contents at the bottom of the cell of a table.
  (k) To give a colour to the background of the entire table.
   (l) To specify the text in a cell of a table to appear in bold letters.
 (m) To place the Caption of the table at the bottom of it.

LONG ANSWER TYPE QUESTIONS

1. Explain the role of HTML in creating web page.

2. Justify the statement: "HTML web pages are always saved as **text** files only".

3. Explain the structure of a HTML document with an example.

4. What is a comment? How do you add comments in a HTML document? Explain with the help of an example.

5. What are the usage of the <frameset> tag?

6. Explain the usage of the TARGET attribute of the <A> tag.

7. What is the significance of the SRC attribute of the <FRAME> tag?

8. Give the output of the following code:

a)
```
<HTML>
<FRAMESET COLS=20%,*>
<FRAME NAME="F1" SRC="a.htm">
<FRAMESET ROWS=50%,*>
<FRAME NAME="F2" SRC=" b.htm">
<FRAME NAME="F3" SRC=" c.htm">
</FRAMESET></FRAMESET>
</HTML>
```

b)
```
<HTML>
<BODY>
<TABLE CELLSPACING="4" CELLPADDING="6" ALIGN="CENTER" BORDER="3">
<CAPTION ALIGN="TOP"><CENTER><B>MARK LIST</B></CENTER></CAPTION>
<TR VALIGN="MIDDLE">
<TH>NAME</TH>
<TH>MARKS</TH>
<TH>PERCENTAGE</TH>
</TR>
<TR VALIGN="MIDDLE">
<TD>MEGHA</TD>
<TD>48/50</TD>
<TD VALIGN="BOTTOM">96%</TD>
</TR>
<TR ALIGN="MIDDLE">
<TD>TARUN</TD>
<TD>15/50</TD>
<TD>28%</TD>
</TR>
<TR ALIGN="CENTER">
<TD>VARUN</TD>
<TD ALIGN="RIGHT" VALIGN="TOP" BGCOLOR="YELLOW">42/50</TD>
<TD>84%</TD>
</TR>
</TABLE>
</BODY>
</HTML>
```

269

Activities

1. Write HTML code to display the names of any 5 subjects on different lines (use <BR> tags). Change the <BR> tags to <P> tags and notice the change. Next try to display the names with a number in front of them. Finally, change the numbers to bullets.

2. Write HTML code to illustrate the usage of nesting of tags. Change the nesting order and see if it makes any difference in the output.

3. Write the HTML code for the following web page:

   List of Subjects

   1. SCIENCE
       (a) BIOLOGY
            (i) BOTANY
            (ii) ZOOLOGY

       (b) CHEMISTRY
            (i) ORGANIC
            (ii) INORGANIC
            (iii) PHYSICAL

       (c) PHYSICS
   2. SOCIAL STUDIES
       (a) GEOGRAPHY
       (b) CIVICS
       (c) HISTORY
       (d) ECONOMICS

4. Play around with the horizontal lines by typing the following code in a notepad file and viewing the output. Try out other variations by changing the size, align and color options.

```
<HTML>
<TITLE> Home Page </TITLE>
<BODY BACKGROUND="bg.gif">
<!- A website created bout famous Cartoon Characters->
What is a Cartoon?<BR>
<HR WIDTH = 50% ALIGN=LEFT SIZE=5 NOSHADE >
<HR SIZE = 8 COLOR = "blue">
<HR WIDTH = 50% ALIGN=RIGHT SIZE=5 COLOR="RED">
<P>A cartoon is any of several forms of illustrations
with varied meanings.
</BODY>
</HTML>
```

5. Write the HTML code to design the following web page:

   ## States of Matter

   There are three states of matter:

   Solids – Example: Pencil

   Liquids – Example: Water

   Gases – Example: Air

   Gases are lighter than solids and liquids.

6. Write the HTML code to design a website (at least two pages) about your favourite Pastime with the following properties:
   (i) Each page should have appropriate titles.
   (ii) The headings should scroll on the page.
   (iii) Insert an appropriate image as a background of the web pages.
   (iv) The heading of the page should be in RED colour and style "Arial". It should be underlined and right aligned.
   (v) The first page should have the information in paragraphs each of which is in different colour and alignment.
   (vi) The second page should contain a list of the famous tourist spots in that area.

7. Write the HTML code to produce the output shown below with the specified numbering.

8. Write the HTML code to generate the output shown below (assume that the name of the file is flower.jpg).
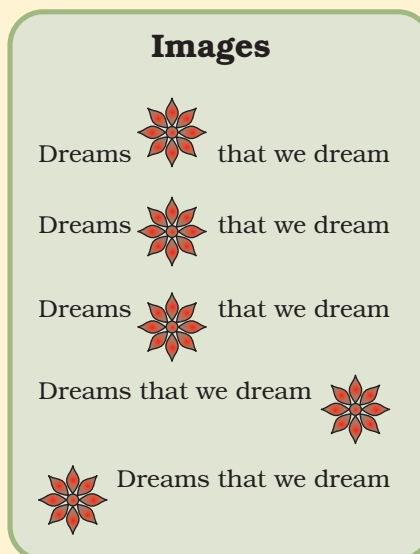
9. Your best friend has created a web page with the url: www.tipod.com/personal/cooldude.html. You now want to link to this web page, when you click on the photograph of your friend which you have placed on your web page. Assume that the name of file containing your friend's photograph is "c:\mypictures\myfriend.jpg".

271

10. Write the HTML code to generate the following output:

Average

Class	Section	Average Marks
XII	A	94.5%
XI	B	91.8%

11. Write the HTML code to generate the following output (assume the names of the images as hill.jpg, sea.jpg and water.jpg respectively).

S No	Place	Image
1.	Hill Stations	
2.	Sea Shores	
3.	Back- waters	

12. Your school's web address is www.littlebuds.com. You want to create a link to this website from your web page. Write the HTML code to do this?

13. Write the HTML code to generate the following page:

> **My School Library**
>
> My school has a big library with lots of books.
>
> Click on any of the categories to see the list of books available :
>
> 1. Fiction
>   - Mystery
>   - Comedy
>   - Drama
>
> 2. Reference Books
>   - Science
>   - Maths
>   - Social Science
>
> 3. Languages
>   - Hindi
>   - English
>   - Sanskrit

Note: Pages linked to (Create the links, assuming that the files already exist):
- Fiction as "fict.html"
- Reference Books as "ref.html"
- Language Books as "lang.html"

14. Write the HTML code to create the following framesets:

(a)

	Second.html
First.html	Third.html
	Fourth.html

(b)

	A.html	
	C.html	
B.html	D.html	E.html

(c)

A.html		C.html
	D.html	
B.html		E.html

CLIENT-SIDE SCRIPTING USING JAVASCRIPT

10

Content creation should not be recondite*. It should not be this bizarre* arcana* that only experts and gold-plated computer science gurus can do.

Brendan Eich
Creator of JavaScript

* recondite- complex, bizarre- strange/ unusual , arcana- deep secret

Objectives

After completing this Chapter, the student will be able to:

- *define JavaScript,*
- *explain the basics of JavaScript,*
- *embed JavaScript code into a HTML document,*
- *compare declare and use of variables,*
- *use variables and literals in expressions,*
- *describe different data types and values,*
- *appreciate the use of branching and loop statements,*
- *perform iteration with **for** loop,*
- *distinguish between **while** and **do...while** loops,*
- *break and continue the loops,*
- *discuss some object manipulation statements and*
- *consider defining and calling of functions.*

Introduction

In Chapter 9 we learnt how to create web pages using HTML. The HTML documents consist of many tags, which tell the browser, how to display the text or graphics. Also, we have learnt to create static web pages; which are not interactive. To make the HTML document interactive and dynamic, there is a need to include some special codes (scripts) along with HTML. One such scripting language is JavaScript.

In this Chapter, we will learn about basics of JavaScript and how to add dynamic effects in web pages using expression, operators, popup boxes, conversion functions, conditional statements, looping statements, object manipulation statements and JavaScript functions.

10.1 ABOUT JAVASCRIPT

JavaScript was created by **Brendan Eich** and it came into existence in September 1995, when Netscape 2.0 (a web browser) was released.

JavaScript was designed with a purpose to make web pages dynamic and more interactive. JavaScript is one of the popular scripting languages having following features :

(a) It can be used for client and server applications.
(b) It is platform independent which means it can run on any operating systems (i.e. Linux, Microsoft Windows, Mac OS X etc.).
(c) JavaScript codes are needed to be embedded or referenced into HTML documents then only it can run on a web browser.
(d) It is an interpreted language.
(e) It is a case-sensitive language and its keywords are in lowercase only.

10.1.1 DIFFERENCE BETWEEN JAVA AND JAVASCRIPT

Some people think that Java and JavaScript are same but both are two completely different languages. Java is a general-purpose object-oriented programming language from Sun Microsystems where as JavaScript is an object-based scripting language. Script refers to short programming statements to perform a task.

10.1.2 VERSIONS OF JAVASCRIPT

Some versions of JavaScript and web browsers are :

JavaScript Version	Web Browsers
JavaScript 1.0	Navigator 2.0, Internet Explorer 3.0
JavaScript 1.3	Navigator 4.06-4.7x, Internet Explorer 4.0
JavaScript 1.5	Navigator 6.0, Mozilla, Internet Explorer 5.5 onwards
JavaScript 1.6	Mozilla Firefox 1.5
JavaScript 1.7	Mozilla Firefox 2.0
JavaScript 1.8	Mozilla Firefox 3.0

Microsoft has released several versions of JavaScript, currently JavaScript version 5.7 is used with Internet Explorer 7.0.

10.2 CLIENT SERVER MODEL

Before stepping ahead, we should know about the Node, Client and Server.

10.2.1 NODE

Node is a component or terminal connected to a network. The components like laptops, PDAs, Internet enabled mobiles etc., can be considered as node in a computer network.

10.2.2 CLIENT

It is a node computer that establishes connection with the server, collects data from the user, sends it to the server, receives information from the server and presents it to the user.

10.2.3 SERVER

In the context of client-server model, server is the counter part of client. It is a computer that serves queries from the client. The programs which respond to the request of clients are known as server applications. The computer designed to run server application is known as server machine. Web server, database server and mail server are some examples of servers.

The Client Server Model is an architecture of computer network where client and server interact by means of a network (*Figure 10.1*). Client gathers data from the user as input and sends request to the server. Server processes the request and sends the requested information to the client. Railway reservation system, online banking and online gaming are examples of client-server model.
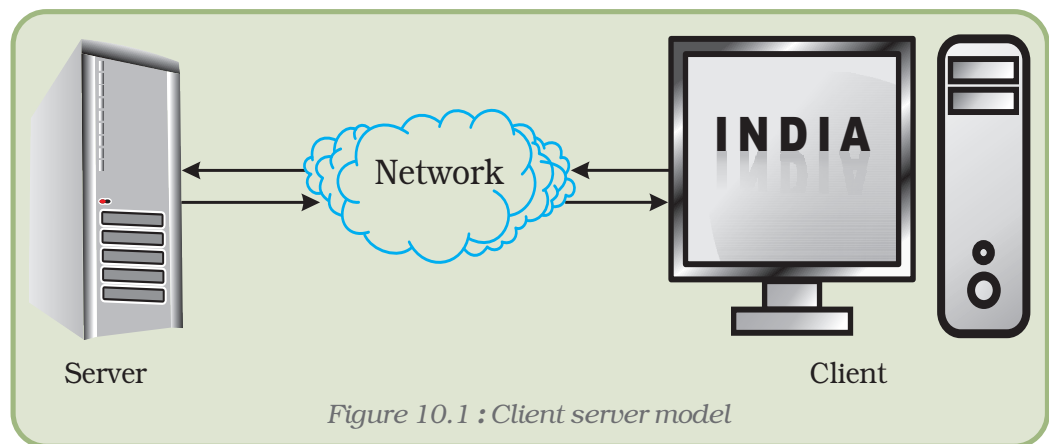
Figure 10.1 : Client server model

10.2.4 CLIENT–SIDE JAVASCRIPT

Client-side JavaScript refers to JavaScript code that gets executed by the web browser on the client machine. JavaScript statements can be embedded in a HTML document that can respond to events such as mouse clicks, form input, and page navigation etc. without any network connection.

10.2.5 SERVER–SIDE JAVASCRIPT

Server-side JavaScript is also known as "LiveWire". Like client-side JavaScript, server-side JavaScript is also embedded within a HTML document. When a query is made by the client, web server executes the script after interpreting it.

10.3 GETTING STARTED WITH JAVASCRIPT

JavaScript is not a full-fledged language and it needs to be embedded within a HTML document. Otherwise, or to specify an external file that contains the JavaScript code we can use word 'script' or 'program' for the JavaScript code.

The most common way to set off a script is to use the HTML <script> and </script> tags in HTML document. We can place our JavaScript code in either the HEAD or BODY section of a HTML document.

The Syntax (General format) is

<SCRIPT [Attributes = ["Value"] ....]>   Indicates starting of JavaScript Code

... JavaScript statement(s);

</SCRIPT>   Indicates ending of JavaScript Code

The following table contains Script attributes, values and their description.
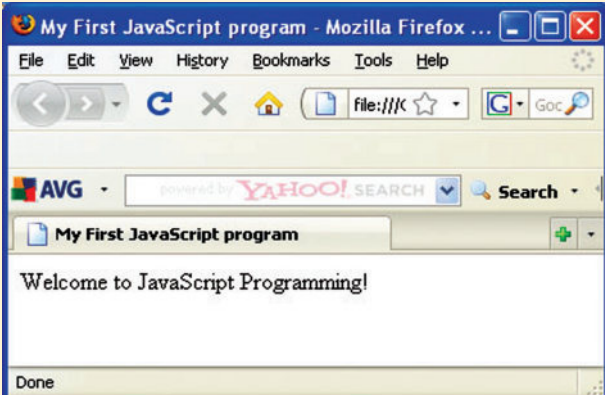
Attribute	Value	Description
Type	text/javascript text/ecmascript text/vbscript	the type of script
Language	Javascript vbscript	name of scripting language
Src	URL	a URL to a file that contains the script

Program 10.1 : First simple JavaScript program using *document.write()*.

```
<HTML>
<HEAD>
<TITLE>My First JavaScript program
</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/javascript">
    document.write("Welcome to ↵
JavaScript Programming!");
</SCRIPT>
</BODY>
</HTML>
```


Note : You notice that the code does not fit into single line. So, we used ↵ to indicate that the code is continued and while inputting it you need not to type it.

To run the above program, type program code given above using any text editor like Notepad, Wordpad and save it as "<file name>.htm" (e.g. abc.htm). Open this file by using any browser application (i.e. Internet Explorer, Mozilla Firefox, Opera etc.).

Tools needed for Writing and Running JavaScript code :

Following tools are needed for working with JavaScript code:

a) **Text Editors:** We can choose any text editor or word processor (i.e. Notepad, Wordpad etc.).

b) **Browser:** Browser interprets JavaScript code and shows the output on browser's document window.

10.3.1 PLACING THE JAVASCRIPT CODE

There are two ways to place the JavaScript code :

1. **Embedded/Inline JavaScript :** JavaScript code can be placed either in the HEAD or in the BODY section of a HTML document.
   a. It is advised to place JavaScript code in HEAD section when it is required to be used more than once.
   b. If the JavaScript code is small in size and used only once, it is advisable to put it in the BODY section of the HTML document.
2. **External JavaScript :** In case, same JavaScript code needs to be used in multiple documents then it is the best approach to place JavaScript code in external files having extension as " **.js**". To do so, we will use *src* attribute in <SCRIPT> tag to indicate the link for the source JavaScript file.

Example : Illustration of the use of external JavaScript code.

```
<HTML>
<HEAD>
<TITLE>Using External JavaScript</TITLE>
</HEAD>
<BODY>
<SCRIPT language="JavaScript" src="abc.js">
</SCRIPT>
<P> The actual JavaScript code exists in external file called
"abc.js". </P>
</BODY>
</HTML>
```

Actual JavaScript file "abc.js"
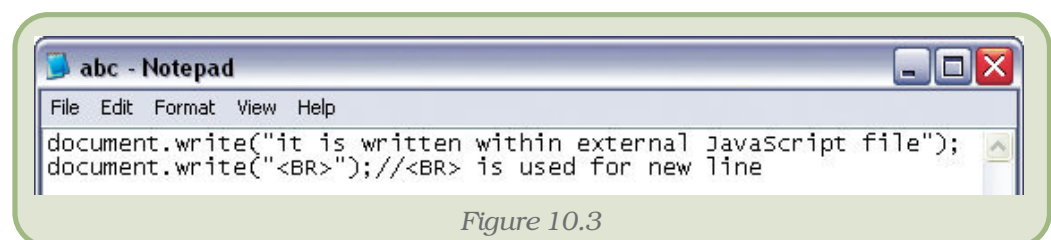
Figure 10.3

Output
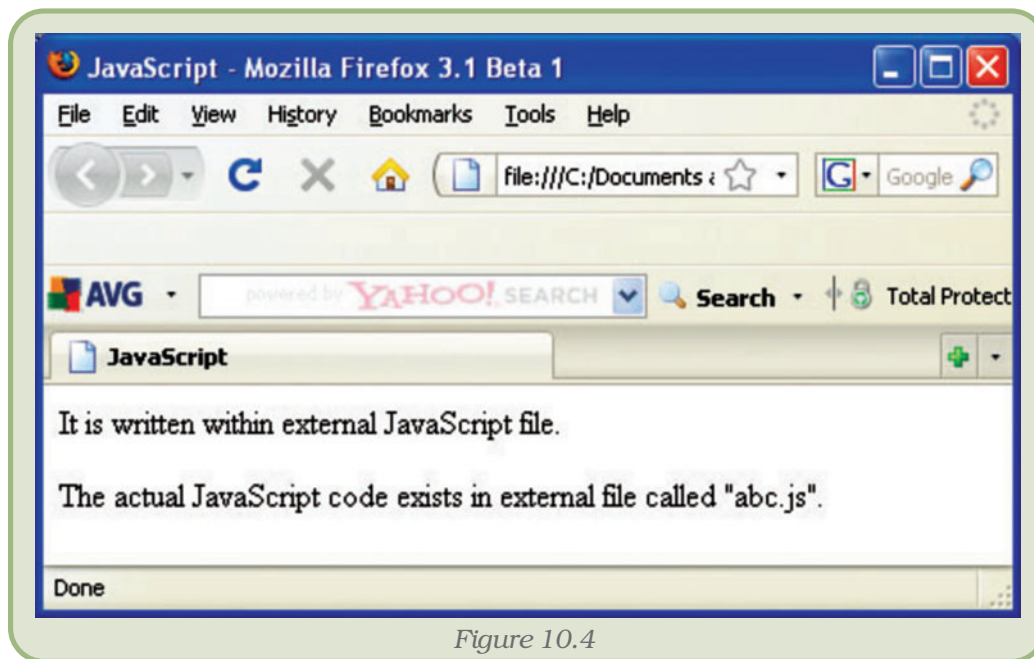
Figure 10.4

10.4 STATEMENTS IN JAVASCRIPT

Statements are the commands or instructions given to the JavaScript interpreter to take some actions as directed. A JavaScript interpreter resides within almost Internet browsers. A collection of statements to accomplish a job, is called as a script or program. The JavaScript statements will be as follows :

```
a = 100;                      // stores value 100 in variable a
b = 200;                      // stores value 200 in variable b
c = a + b;                    // stores the sum of a and b in
                              // variable c
document.write
("Sum of A and B : ");        // displays the string
document.write(c);            // displays the value of c
```

In JavaScript, semicolon(;) is used to end a statement but if two statements are written in separate lines then semicolon can be omitted. Some valid statements are :

(i) p=10
   q=20
(ii) x=12; y=25   // semicolon(;) separating two statements.

Some invalid statements are :

x=12 y=25 // statements within the same line not separated by semicolon (;)

10.4.1 COMMENTS

Comments are the statements that are always ignored by the interpreter. They are used to give remarks to the statement making it more readable and understandable to other programmers. There are two types of comments :

- Single line comment using double-slash (//).

- Multiple lines comment using /* and */ .

For example :

// This is a single-line comment.

/* This is a multiple-line comment.

   It can be of any length.   */

10.4.2 LITERALS

Literals refer to the constant values, which are used directly in JavaScript code. For example:

a=10;

b=5.7;

document.write("Welcome");

In above statements *10*, *5.7*, *"Welcome"* are literals.

10.4.3 IDENTIFIERS

Identifiers refer to the name of variables, functions, arrays, etc. created by the programmer. It may be any sequence of characters in uppercase and lowercase letters including numbers or underscore and dollar sign. An identifier must not begin with a number and cannot have same name as any of the keywords of the JavaScript.

Some valid identifiers are :

RollNo

bus_fee

_vp

$amt

Some invalid identifiers are :

to day   // Space is not allowed

17nov   // must not begin with a number

%age   // no special character is allowed

10.4.4 RESERVED WORDS OR KEYWORDS

Reserved words are used to give instructions to the JavaScript interpreter and every reserved word has a specific meaning. These cannot be used as identifiers in the program. This means, we cannot use reserved words as names for variables, arrays, objects, functions and so on. These words are also known as "Keywords". A list of reserved words in JavaScript is given in Appendix 10.1.

10.4.5 VARIABLES

A variable is an identifier that can store values. These values can be changed during the execution of script. Once a value is stored in a variable it can be accessed using the variable name. Variable declaration is not compulsory, though it is a good practice to use variable declaration. Keyword **var** is used to declare a variable.

Syntax

var *var-name* [= *value*] [..., *var-name* [= *value*]]

Example

>   var name = "Sachin";   // Here 'name' is variable
>
>   document.write(name);   // Prints Sachin
>
>   A JavaScript variable can hold a value of any data type. For example :
>
>   i = 7;
>
>   document.write(i);   // prints 7
>
>   i = "seven";   // JavaScript allows to assign string values
>
>   document.write(i);   // prints seven
>
>   Some valid examples of variable declaration:
>
>   var cost;
>
>   var num, cust_no = 0;
>
>   var amount = 2000;

Naming Conventions

We should use meaningful name for a variable. A variable name must start with a letter, underscore (_), or dollar sign ($). The subsequent characters can be the digits (0-9). JavaScript is *case sensitive,* so the variable name my_school is not the same as My_School.

Some valid variable names

>   f_name
>
>   India123

281

_sumof

Some invalid variable names

10_numbers   - *must not begin with any number.*

rate%   - *'%' is not a valid character.*

my name   - Space is not allowed.

Program 10.2 : To find the sum of two numbers using ***var***.

```
<HTML>
<HEAD>
<TITLE>Sum of two numbers</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/javascript">
var a = 100;
var b = 500;
var c = a + b;
document.write ("Sum of a & b ↵
: "  + c );
</SCRIPT>
</BODY>
</HTML>
```
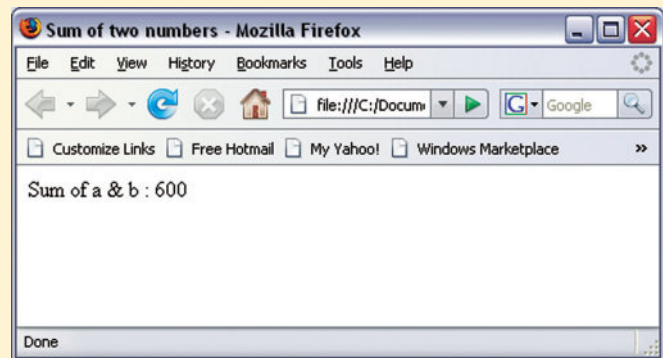

Figure 10.5

10.5 DATA TYPES

JavaScript supports three basic data types – number, string, boolean and two composite data types – arrays and objects.

10.5.1 NUMBER

The number variable holds any type of number, either an integer or a real number. Some examples of numbers are:

29, -43, 3.40, 3.4323

10.5.2 STRING

A string is a collection of letters, digits, punctuation characters, and so on. A string literal is enclosed within single quotes or double quotes ('or "). Examples of string literals are:

'welcome', "7.86" , "wouldn't you exit now", ' country="India" '

JavaScript also allows us to use Escape Sequence within string literals. The escape sequence starts with a backslash (\), followed by another character. This backslash tells browser to represent a special action or character representation. For example, \" is an escape sequence that represents a double quote (").

282

Escape Sequence	Action/ Character Represented
\b	Backspace
\n	New line
\r	Carriage return
\t	Tab
\'	Single quote (')
\"	Double quote (")
\\	Backslash (\)

Example :

```
document.write ("Abhinav said, \"Earth doesn\'t revolve round↵
the sun\". But teacher corrected him.");
```

Here, two types of escape characters are used \" and \' in this example.

Output

Abhinav said, "Earth doesn't revolve round the sun". But teacher corrected him.

10.5.3 BOOLEAN VALUES

A *boolean* variable can store only two possible values either true or false. Internally it is stored as 1 for *true* and 0 for *false*. It is used to get the output of conditions, whether a condition results in true or false.

Example

x == 100;   // results true if x=100 otherwise false.

10.5.4 ARRAYS

An array is a collection of data values of same types having a common name. Each data element in array is referenced by its position in the array also called its index number. Individual array elements can be referenced by the array name followed by the pair of square brackets having its index number. The index number starts with zero in JavaScript i.e. the first element in JavaScript has it index value as 0, second has its index value as 1 and so on. An array can be declared in any of the following ways :

    var a = new a( );

```
var x = [ ];

var m = [2,4,"sun"];
```

An array is initialised with the specified values as its elements, and its length is set to the number of arguments specified.

Example This creates an array name games with three elements.

```
games = ["Hockey", "Cricket", "Football"];
```

We can also store different types of values in an array.

For example :

```
var arr = new Array();          // creation of an array

arr[0] ="JAVASCIPT";            // stores String literal at index 0

arr[1] = 49.5;                  // stores real number at index 1

arr[2] = true;                  // stores Boolean value
```

10.5.5 NULL VALUE

JavaScript supports a special data type known as ***null*** that indicates "no value or blank". Note that ***null*** is not equal to 0.

Example

```
var distance = new object();

distance = null;
```

10.6 OBJECTS

JavaScript is an object based scripting language. It allows us to define our own objects and make our own variable types. It also offers a set of predefined objects. The tables, forms, buttons, images, or links on our web page are examples of objects. The values associated with object are properties and the actions that can perform on objects are methods or behaviour. Property associated to an object can be accessed as follows:

ObjectName.PropertyName

Now we will study, some of the predefined objects in JavaScript.

10.6.1 DOCUMENT OBJECT

The Document object is one of the parts of the Window object. It can be accessed through the **window.document** property. The ***document*** object represents a HTML document and it allows one to access all the elements in a HTML document. For example: title of current document can be accessed by **document.title** property.

Some of the common properties of document object are:

Properties	Purposes
Title	returns/ sets title of the current document.
bgColor	returns/ sets the background color of the current document.
fgColor	returns/ sets the text color of the current document.
linkColor	returns/ sets the color of hyperlinks in the document.
alinkColor	returns/ sets the color of active links in the document.
vlinkColor	returns/ sets the color of visited hyperlinks.
height	returns the height of the current document.
width	returns the width of the current document.
Forms	returns a list of the FORM elements within the current document.
Images	returns a list of the images in the current document.
URL	returns a string containing the URL of the current document.
Location	to load another URL in current document window.

Methods	Purposes
open()	Opens a document for writing.
write()	Writes string/data to a document.
writeln()	Writes string/data followed by a newline character to a document.
close()	Closes a document stream for writing.

Program 10.3 : To illustrate the properties and methods of the document object.

```
<HTML>
<HEAD>
<TITLE>Document Properties</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/javascript">
document.fgColor = "green"; // sets text color
document.bgColor = "yellow";      // background color
document.title = "Chakde India";  // change title
document.linkColor = "navy";      // hyperlinks color
document.alinkColor = "red";      // active links
document.vlinkColor = "lime";     // visited hyperlinks
document.write("<BR>Do you want to play for India?");
document.write("<BR> <A href='yes.html'> Yes </A>");
document.writeln("<BR> <A href='no.html'>No</A><BR>");
document.write("<BR>Title of current document: " + document.title);
```

285

```
document.write("<BR>Height of current document: " + document.height);
document.write("<BR>Width of current document: " + document.width);
document.write("<BR> URL of current document is: " + document.URL);
//Use of document.links to list of all the hyperlinks
document.write("<BR><B>The List of Links in current document</B>");
var links = document.links;
for(var i = 0; i < links.length; i++)
{
document.write("<BR>"+ document.links[i]);
}
</SCRIPT>
</BODY>
</HTML>
```
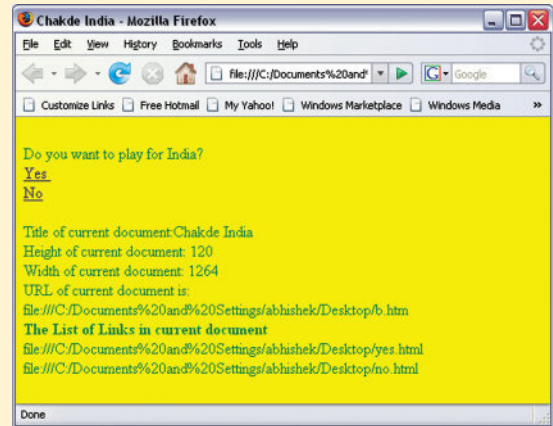

Figure 10.6

10.6.2 DATE OBJECT

This object is used to set and manipulate date and time. JavaScript dates are stored as the number of milliseconds since midnight, January 1, 1970. This date is called the epoch. Dates before 1970 are represented by negative numbers. A date object can be created by using the *new* keyword with Date().

Syntax

newDate()

new Date(milliseconds)

new Date(dateString)

new Date(yr_num, mo_num, day_num

[, hr_num, min_num, sec_num, ms_num])

Parameters

Milliseconds	Milliseconds since 1 January 1970 00:00:00.
dateString	Date String. e.g. "October 5, 2007"
yr_num, mo_num,day_num	Year (e.g. 2007)Month (Value 0-11, 0 for January and 11 for December), Day (1-31)
hr_num, min_num,sec_num, ms_num	Values for Hour, Minutes, Second and milliseconds

Different examples of using a date()

```
today = new Date();
dob = new Date("October 5, 2007 12:50:00");
doj = new Date(2007,10,5);
bday = new Date(2007,10,5,12,50,0);
```

Methods to read date values

We can use the **get** methods to get values from a Date object. Here are some get methods that returns some value according to local time:

getDate( )	Returns the day of the month
getDay( )	Returns the day of the week
getFullYear( )	Returns the full year
getHours( )	Returns the hour
getMinutes( )	Returns the minutes
getMonth( )	Returns the month
getSeconds( )	Returns the seconds
getTime( )	Returns the numeric value corresponding to the time
getYear( )	Returns the year

Program 10.4 : A simple JavaScript program that displays today's date and current time.

```
<HTML>
<HEAD>
    <TITLE>Displaying Time</TITLE>
</HEAD>
<BODY>
    <CENTER>
        <H1>Today's Date and Current Time</H1>
    </CENTER>
    <SCRIPT type="text/javascript">
        var today = new Date();
        document.write("<H2>"); // JavaScript allows the use
        document.write(today);  // of HTML formatting tag
        document.write("</H2>"); // with document.write
    </SCRIPT>
</BODY>
</HTML>
```
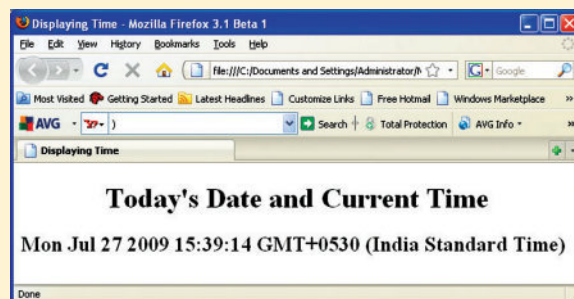

Figure 10.7

287

10.6.3 MATH OBJECT

This object contains methods and constants to carry more complex mathematical operations. This object cannot be instantiated like other objects. All properties and methods of **Math** are static. We can refer to the constant p as **Math.PI** and the sine function as **Math.sin(x)**, where **x** is the method's argument.

Properties	Description
Math.PI	Returns the value of p
Math.E	Euler's constant and the base of natural logarithms.
Math.LN2	Natural logarithm of 2.
Math.LN10	Natural logarithm of 10, approximately 2.302.
SQRT1_2	Square root of ½.
SQRT2	Square root of 2.

Methods	+Description
pow(x, p)	Returns X^p
abs(x)	Returns absolute value of x.
exp(x)	Returns e^x
log(x)	Returns the natural logarithm of x.
sqrt(x)	Returns the square root of x.
random()	Returns a random number between 0 and 1.
ceil(x)	Returns the smallest integer greater than or equal to x.
floor(x)	Returns the largest integer less than or equal to x.
min(x, y)	Returns the lesser of x and y.
max(x, y)	Returns the larger of x and y.
round(x)	Rounds x up or down to the nearest integer.
sin(x)	Returns the sin of x, where x is in radians.
cos(x)	Returns the cosine of x, where x is in radians.
tan(x)	Returns the tan of x, where x is in radians.

Example : To illustrate the properties and methods of the Math object.

```
<HTML>
<HEAD>
<TITLE>Math Object</TITLE>
</HEAD>
<BODY>
<SCRIPT type="text/JavaScript">
document.write("Value of PI :"+Math.PI + "<BR>");
document.write("Random value:"+Math.random()+"<BR>");
document.write("Rounded value of 0.69 :"+ ↵
Math.round(0.69)+"<br>");
document.write("Value of 5 <SUP>2</SUP> :"+ ↵
```

```
Math.pow(5,2) + "<br>");
document.write("Square root of 2 :"+Math.SQRT2 );
</SCRIPT>
</BODY>
</HTML>
```


Figure 10.8

10.7 EXPRESSIONS AND OPERATORS

An expression is a combination of operators operands that can be evaluated. It may also include function calls which return values.

Examples

```
x = 7.5                    // a numeric literal
"Hello India!"             // a string literal
false                      // a Boolean literal
{feet:10, inches:5}        // an object literal
[2,5,6,3,5,7]              // an array literal
v= m + n;                  // the variable v
tot                        // the variable tot
```

10.7.1 ARITHMETIC OPERATORS

These are used to perform arithmetic/mathematical operations like subtraction, division, multiplication etc. Arithmetic operators work on one or more numerical values (either literals or variables) and return a single numerical value. The basic arithmetic operators are:

+	(Addition)	**-**	(Subtraction)
*****	(Multiplication)	**/**	(Division)
%	(Modulus)	**++**	(Increment by 1)
—	(Decrement by 1)		

Examples

```
var s = 10 + 20;           // result: s=30
var h = 50 * 4;            // result: h = 200
```

289

```
var d = 100 / 4;              // result: d = 25
var r = 72 % 14;             // result: r=2
```

Increment and decrement operators

These operators are used for increasing or decreasing the value of a variable by 1. Calculations performed using these operators are very fast.

Example

```
var a = 15;
a++;                         // result: a = 16
var b = 20;
b−;                          // result: b = 19
```

10.7.2 ASSIGNMENT OPERATORS

It assigns the value of its right operand to its left operand. This operator is represented by equal sign(=).

Example

```
x = 100;         // This statement assigns the value 100 to x.
```

JavaScript also supports shorthand operator for standard operations. The shorthand operator with example :

Shorthand operator	Example	is equivalent to
+ =	a + = b	a = a + b
– =	a – = b	a = a – b
* =	a * = b	a = a * b
/ =	a / = b	a = a / b
% =	a % = b	a = a % b

10.7.3 RELATIONAL (COMPARISON) OPERATORS

Relational Operators are some symbols which return a Boolean value true or false after evaluating the condition. For example x > y; returns a value true is value of variable x is greater than variable y.

Basic JavaScript comparison operators are given in the table below :

Operator	Description	Example
= =	is equal to	4 = = 8 returns false
! =	is not equal to	4 ! = 8 returns true
>	is greater than	8 > 4 returns true
<	is less than	8 > 4 returns false
< =	is less than or equal to	8 < = 4 returns false
> =	is greater than or equal to	8 > = 4 returns true

Relational operators are functional for strings as well. The comparison takes place in alphabetical order. This alphabetical order is based on ASCII number. For example :

Statement	Output
"zero" < "one"	// false
"Zero" < "one"	// true
10 < 5	// false, numeric comparison.
"10" < "5"	// true, string comparison.
"10" < 5	// false, numeric comparison;
"Ten" < 5	// Error occurs, "Ten" can not be
	// converted into a number

10.7.4 LOGICAL OPERATORS

Logical operators are used for combining two or more conditions. JavaScript has following three logical operators :

Operator	Description with Example
&& (AND)	returns *true* if both operands are *true* else it return *false*.
|| (OR)	returns *false* if both operands are *false* else it returns *true*.
! (NOT)	returns *true* if the operand is *false* and false if operand is *true*.

10.7.5 CONCATENATION OPERATOR

The + operator concatenates two string operands. The + operator gives priority to string operands over numeric operands It works from left to right. The results depend on the order in which operations are performed. For example :

Statement	Output
"Good" + "Morning"	"GoodMorning"
"5" + "10"	"510"
"Lucky" + 7	"Lucky7"
4 + 7 + "Delhi"	"11Delhi"
"Mumbai" + 0 +0+ 7	"Mumbai007"

10.7.6 SPECIAL OPERATORS

Conditional Operator (? :)

The conditional operator is a special JavaScript operator that takes three operands. Hence, it is also called ternary operator. A conditional operator assigns a value to a variable based on the condition.

Syntax

```
var_name = (condition) ? v_1 : v_2
```

If (condition) is true, the value v_1 is assigned to the variable, otherwise, it assigns the value v_2 to the variable.

For example

```
status = (age >= 18) ? "adult" : "minor"
```

This statement assigns the value "adult" to the variable status if age is eighteen or more. Otherwise, it assigns the value "minor" to status.

New

new operator is used to create an instance and allocate memory to a user-defined or predefined object types.

Syntax

ObjectName = new objectType (param1 [,param2] ...[,paramN])

Example

```
d = new Date(); // date assigns to object d
r = new rectangle(4, 5, 7, 8);
```

Delete

The *delete* operator de-allocates (releases) the memory space that was allocated using the new operator by deleting an object, an object's property or an element from an array.

The syntax is

```
delete object_name
delete object_name.property
delete array_name[index]
```

delete operator can be used to delete variables declared implicitly but not those declared with the *var* statement. The delete operator returns true if the operation is possible; it returns false if the operation is not possible.

```
a=12
var j= 63
myobj=new Number()
myobj.h=55          // create property h
delete x            /* returns true (x is declared implicitly,
                    without using var)*/
delete y            /* returns false ( y is declared
                    explicitly using var) */
delete Math.PI      /* returns false (cannot delete predefined
                    properties)*/
delete myobj.h      /* returns true (can delete user-defined
                    properties)*/
```

```
delete myobj              /* returns true (can delete if declared
                             implicitly) */
```

When we delete an array element, the array length will not be affected. For example, if we delete a[3], then a[4] still remains a[4] and a[3] will be undefined. When the *delete* operator removes an array element, that element is no longer in the array.

this

JavaScript supports **this** operator. The word **this** refers to the current object. It is like a pointer to the current object.

The syntax is

```
this[.propertyName]
```

Example

Use of *this* operator to validate the age. Here input is provided through the text box.

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
    function validate(obj, min_age, max_age)
{
   if ((obj.value < min_age) || (obj.value > max_age))
      alert("Invalid age for the Job!!!");
}
</script>
</HEAD>
<BODY>
<B>Enter the age (between 18 and 40):</B>
<INPUT TYPE = "text" NAME = "age" SIZE = 2
onChange="validate(this, 18, 40)">
</BODY>
</HTML>
```

In this example we called validate() function by onChange event handler. Here, this operator is used to pass current object (viz. text box).
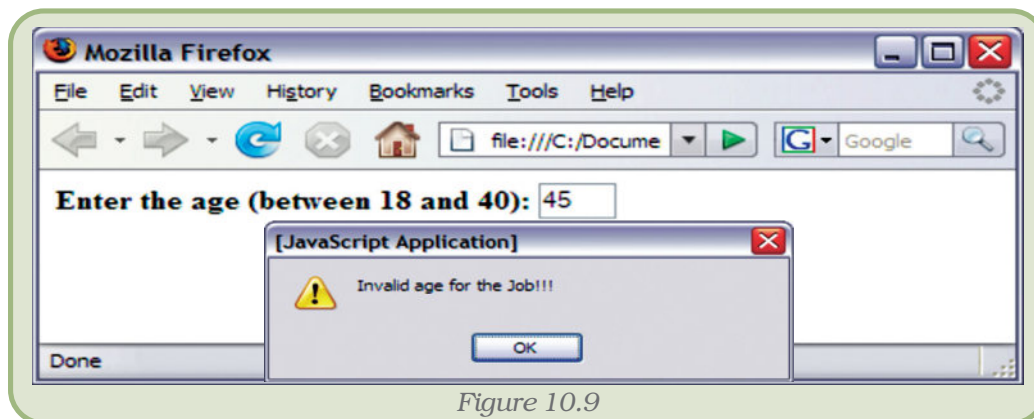

Figure 10.9

293

10.7.7 OPERATOR PRECEDENCE

Operators are evaluated in a predefined order of precedence. The following table shows operators from highest priority to lowest priority :

Table 10.1 : Operator Precedence

Operator	Description	Priority
* / %	Multiplication Division Modulus	**Highest**
+ –	Addition Subtraction	
< <= > >=	Less than Less than equal to Greater than Greater than equal to	
== !=	Equality Not equality	
&&	Logical AND	
||	Logical OR	
?:	Conditional	
= += –= *= /= %=	Assignment Operators	
,	Comma	**Lowest**

10.8 JAVASCRIPT POPUP BOXES (DIALOG BOXES)

In JavaScript, three kinds of popup boxes – Alert box, Confirm box, and Prompt box can be created using three methods of window object.

10.8.1 ALERT BOX

Alert( ) method of window object creates a small dialog box with a short text message and "OK" command button called alert box. Alert box contains an icon indicating a warning.

Syntax

```
[window].alert("Text to be displayed on the popup box");
```
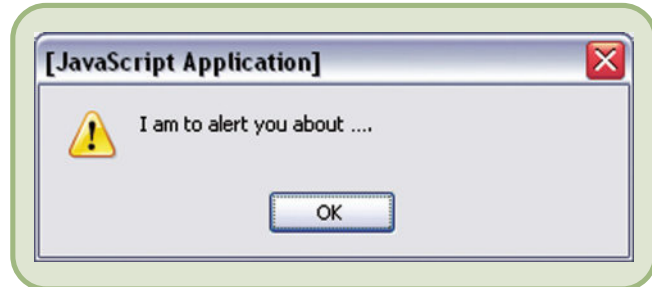
The word *window* is optional.

Example

```
window.alert("I am to alert you about ....");
or
alert("I am to alert you about ....");
```

Output

An alert box is used if we want to display some information to the user. When an alert box appears, the user needs to click "OK" button to proceed.

10.8.2 CONFIRM BOX

Confirm box is used if we want the user to verify and confirm the information. The user will have to click either "OK" or "Cancel" buttons.

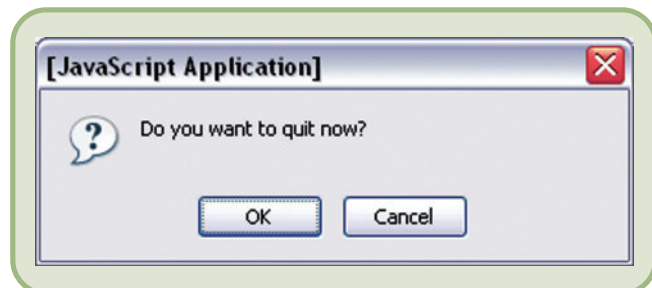### Syntax

```
[window].confirm("Text to be confirmed");
```

Example

```
confirm("Do you want to quit now?");
```

Output

Confirm box returns a Boolean value. If the user clicks on "OK", it returns true. If the user clicks on "Cancel", it returns false.

10.8.3 PROMPT BOX

Prompt box allows getting input from the user. We can specify the default text for the text field. The information submitted by the user from *prompt( )* can be stored in a variable.
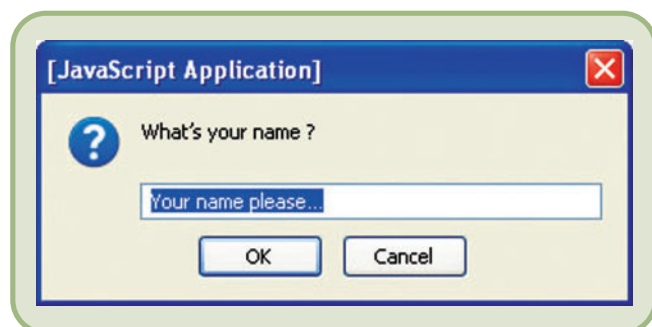
Syntax

```
prompt("Message" [, "default value in the text field"]);
```

Example

```
var name = prompt("What's your ↵
name? ", "Your name please...");
```

Output

A prompt box returns input string value when the user clicks "OK". If the user clicks "Cancel", it returns null value.

295

10.9. BLOCK STATEMENT

Two or more statements can be combined to form a single block statement using a set of curly brackets.

The Syntax is

```
{
    statement_1
    statement_2
    .
    .
    statement_n
}
e.g.   If (z > y)
{    x=10;
y=20;   }
```

10.10 BRANCHING AND LOOPING STATEMENTS

JavaScript allows to select among alternative ways or to repeat the execution of a statement or block of statements. JavaScript supports some conditional statements for the branching. A conditional statement is a statement that we can use to execute a part of code based on a condition or to do something else if the condition is not met.

Looping is repeating execution of a set of statements for a number of times.

10.10.1 BRANCHING (CONDITIONAL) STATEMENTS

Branching With *If* Statements

An *if* statement is used to execute a statement or a block of statements on based of logical expression (condition). There are three different forms :

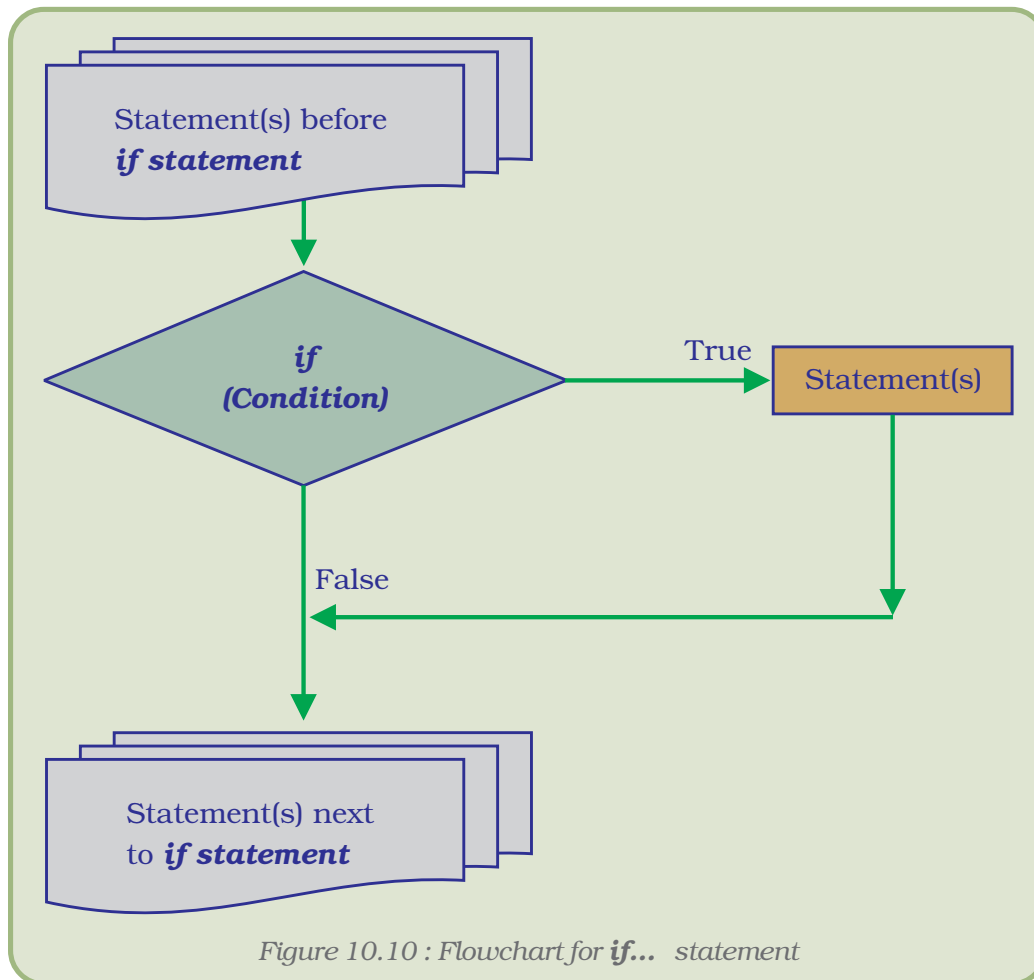 - if … statement (simple *if* statement).
  - if … else statement.
  - If .. else if .. else statement (*else if* ladder statement)

if … statement

The '*if*' is the simplest decision making statement. This statement is used to execute statement(s) only if a specified condition is true.

The Syntax is

```
if ( condition )
{
.. statement(s) to be executed if (condition)  is true...
}
```

*Figure 10.10 : Flowchart for **if...** statement*

In the above flowchart, statement(s) is/are executed only when logical expression is true. Otherwise, the statement following 'if statement' will be executed next.

Example : A JavaScript program that displays 'Good Morning India!' if and only if time is less than 12 hours on web page otherwise page will remain blank.

```
<HTML>
<HEAD>
    <TITLE>if statement</TITLE>
</HEAD>
<BODY>
<script type = "text/javascript">
var d = new Date();
var time = d.getHours();  // time stores hours
if (time < 12)
{
    document.write("Good Morning India!");
}
</script>
```

297

```
</BODY>
</HTML>
```

This HTML document displays the message "Good Morning India!" if your system time is less then 12 Hrs. Other wise you will find the page blank.

Output
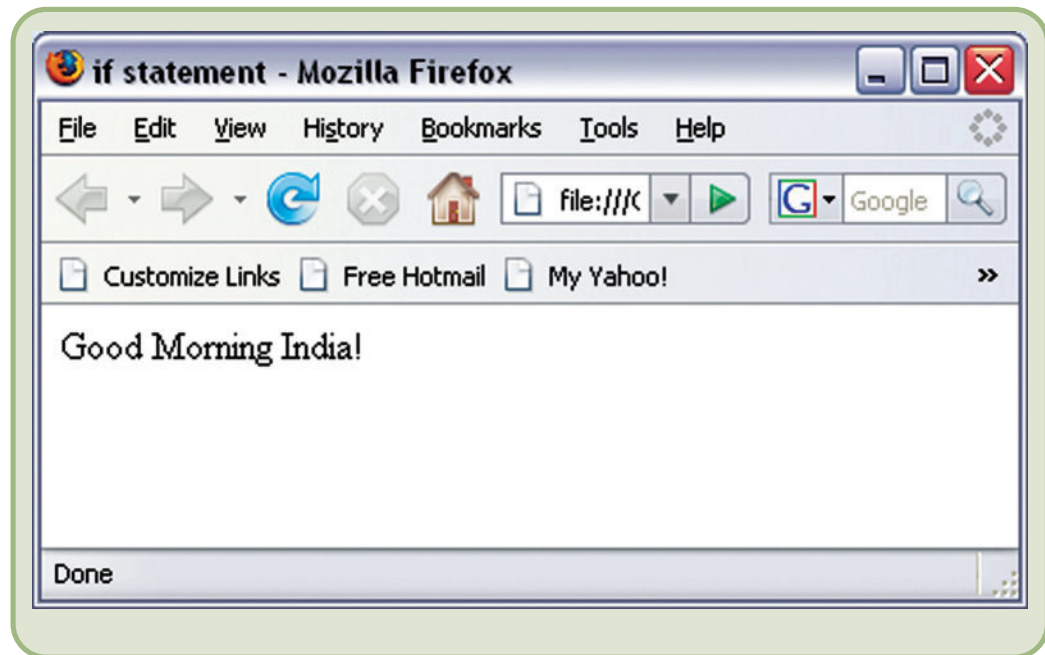
if ... else ... statement

This statement is an extension of the simple if statement. It permits one of two statement or group of statements depending upon the logical test.

The Syntax is

> if (*condition*)
>
> {
>
>   *True statement(s)...*
>
> }
>
> else
>
> {
>
>   *False statement(s)...*
>
> }

If the logical expression (condition) is true, the true statement(s) will be executed. Otherwise, the false statement(s) will be executed.

*Figure 10.11: Flowchart for **if ... else ...** statement*
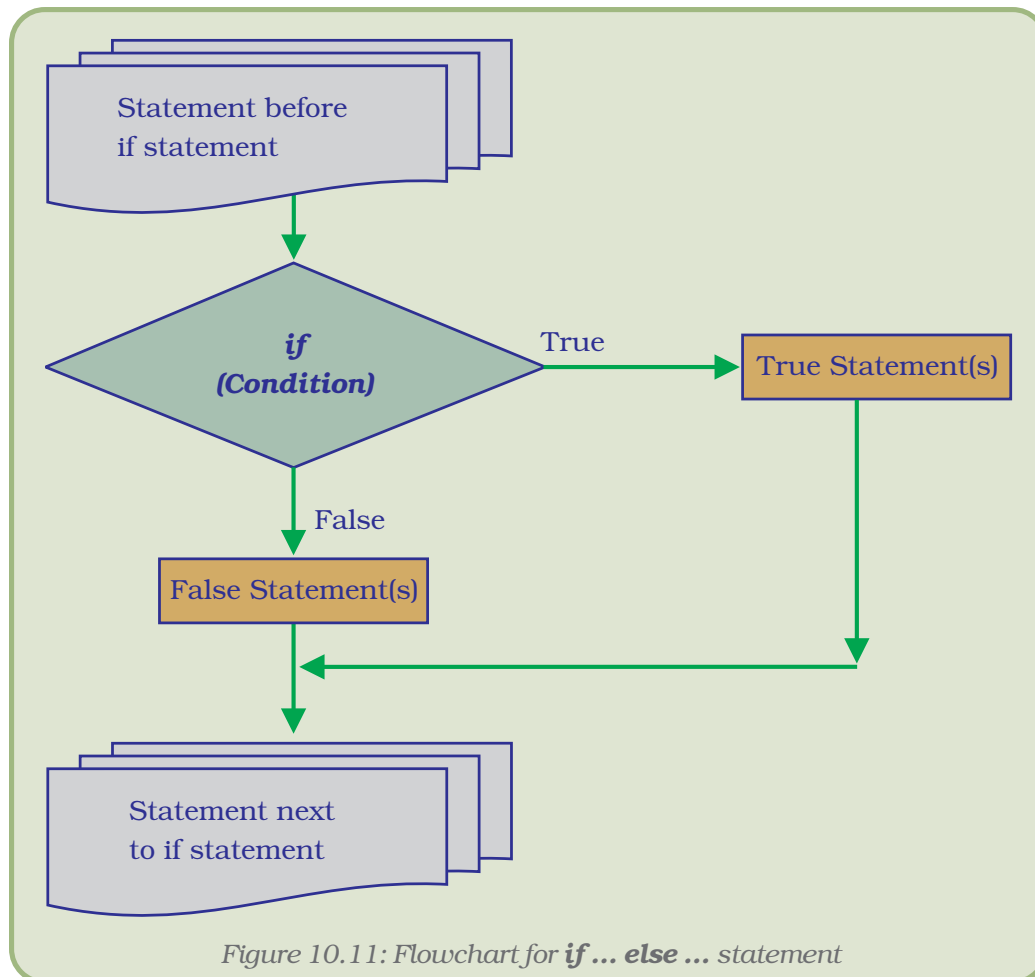
Example : A JavaScript program that displays 'Good Morning India!' if time is less than 12 hours otherwise it shows 'Good Day India!' on the document.

```
<HTML>
<HEAD>
    <TITLE>if else statement</TITLE>
</HEAD>
<BODY>
<script type = "text/JavaScript">
var d = new Date();
var time = d.getHours();
if (time < 12)
{
    document.write("Good Morning India!");
}
else
{
    document.write("Good Day India!");
}
</SCRIPT>
```

299

```
</BODY>
</HTML>
```

This HTML document displays the message "Good Morning India!" if the system time is less then 12 hours. Other wise it display the message "Good Day India!".

Output
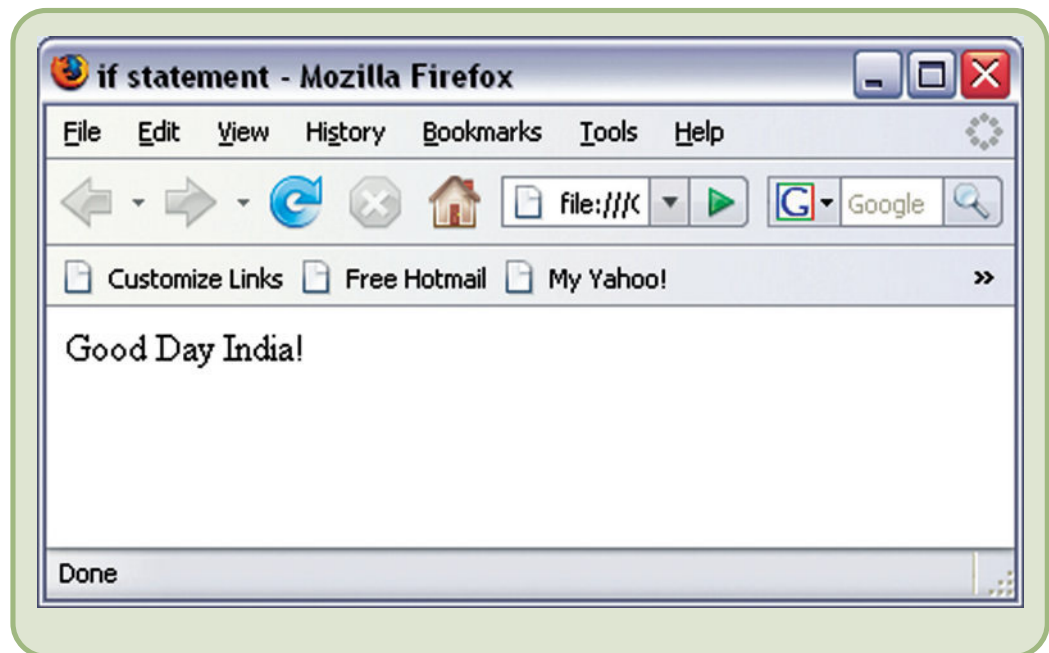
If...else if...else Statement

The if....else if...else statement is further an extension of the if... else... statement. This statement is useful for selecting one of many sets of statements to execute.

The Syntax is

if (*condition1*)

{

    code to be executed if condition1 is true

}

else if (*condition2*)

{

    code to be executed if condition2 is true

}

.

.

else
{
    code to be executed if any of the conditions is not true
}

*Figure 10.12 : Flowchart for **if ... else if ... else** statement*
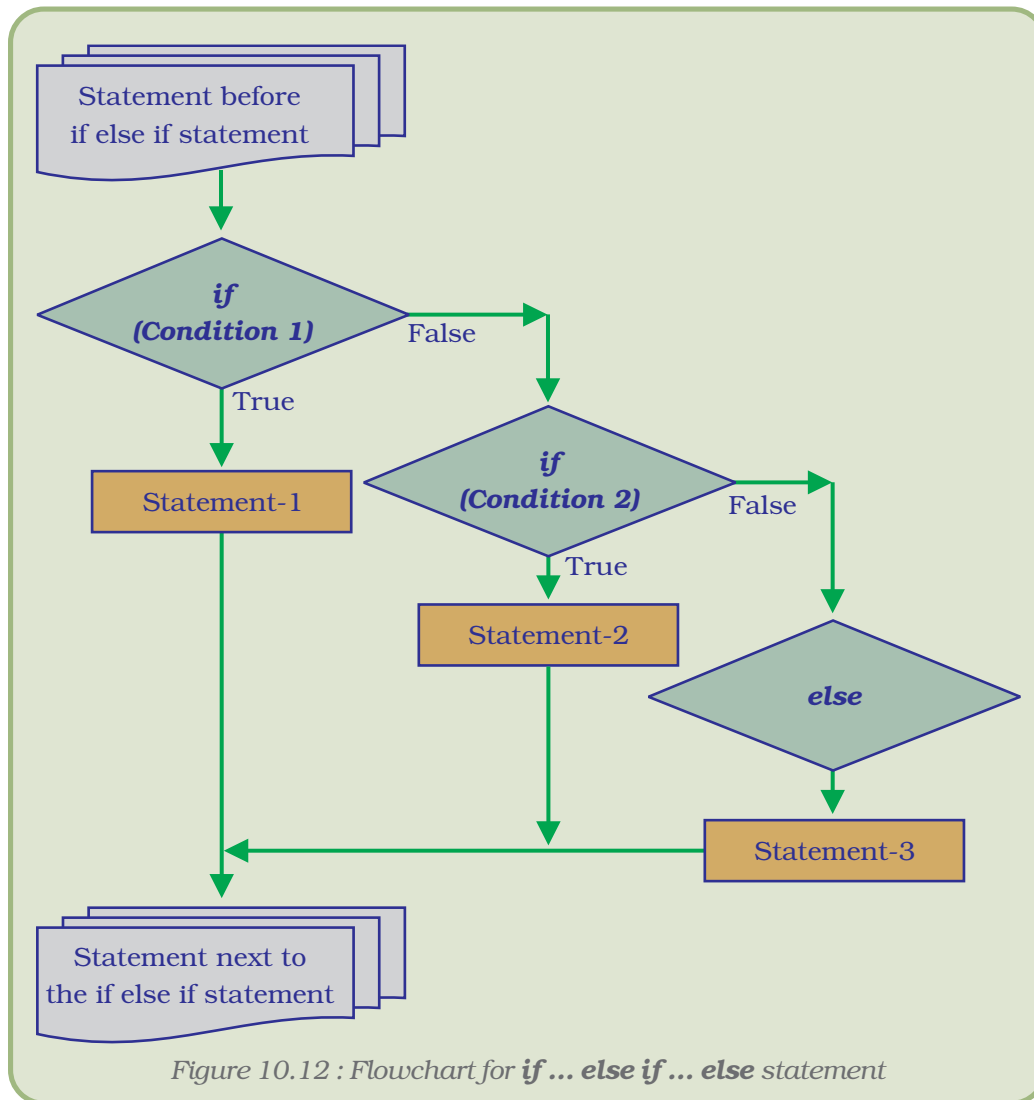
Example : Write a program to check whether a number input through prompt box is Zero, Odd or Even.

```
<HTML>
<HEAD>
    <TITLE>Odd, Even or Zero</TITLE>
</HEAD>
<BODY>
<SCRIPT type = "text/JavaScript">
    var n = prompt("Enter your number:","Type your number↵
here");
```

```
n = parseInt(n);          //converts string into number
if (n == 0)
      alert("The number is zero!");
else if (n%2)
      alert("The number is odd!");
else

          alert("The  number  is
even!");
</SCRIPT>
</BODY>
</HTML>
```

Output

Selection with switch statement

A switch statement is used to execute different statement based on different conditions. It provides a better alternative than a long series of if... else if ... statements.

The Syntax is

> switch (*expression*)
>
> {
>
>   case *label1* :   //executes when value of exp. evaluates to label
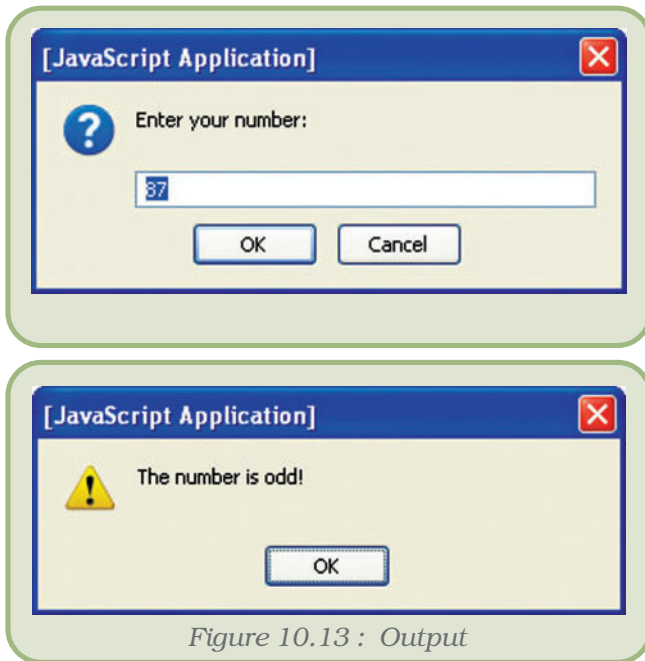
Figure 10.13 : Output

> *statements*;
>
> break;
>
> case *label2* :
>
> *statements*;
>
>   break;
>
> …
>
> default : *statements*;   //executes when non of the above labels
>
>   //matches the result of expression
>
> }

Program 10.5 : To compute the day of week (in words) while you input the date within prompt dialog box.

```
<HTML>
<HEAD>
    <TITLE>Switch statement</TITLE>
</HEAD>
<BODY>
```

```
<script type="text/JavaScript">
// Enter date of birth to know the day of week on that day.
var d=new Date(prompt("Enter your Date of Birth
(e.g. November 17, 2002)", "Month DD, YYYY"))
dy=d.getDay()
switch (dy)
{
case 0:
    document.write("It was <b>Sunday</b> on that day.")
    break
case 1:
    document.write("It was <b>Monday</b> on that day.")
    break
case 2:
    document.write("It was <b>Tuesday</b> on that day.")
    break
case 3:
    document.write("It was <b>Wednesday</b> on that day.")
    break
case 4:
    document.write("It was <b>Thursday</b> on that day.")
    break
case 5:
    document.write("It was <b>Friday</b> on that day.")
    break
case 6:
    document.write("It was <b>Saturday</b> on that day.")
    break
default:
    document.write("Please  input  a
valid Date in?
prescribed format !!!")
}
</script>
</BODY>
</HTML>
```

The value of this expression is then compared with the values for each case in the switch structure. If there is a match, the block of code associated with that case is executed. If no case is matched, the statement in default will be executed. Use **break** to prevent the code from running into the next case automatically. There is no need to use break within the default case.

When you press OK button, we will get the output (Figure 10.14) :

10.10.2 LOOP STATEMENTS

Loop statements are the primary mechanism for telling a JavaScript interpreter to execute statements again and again until a specified
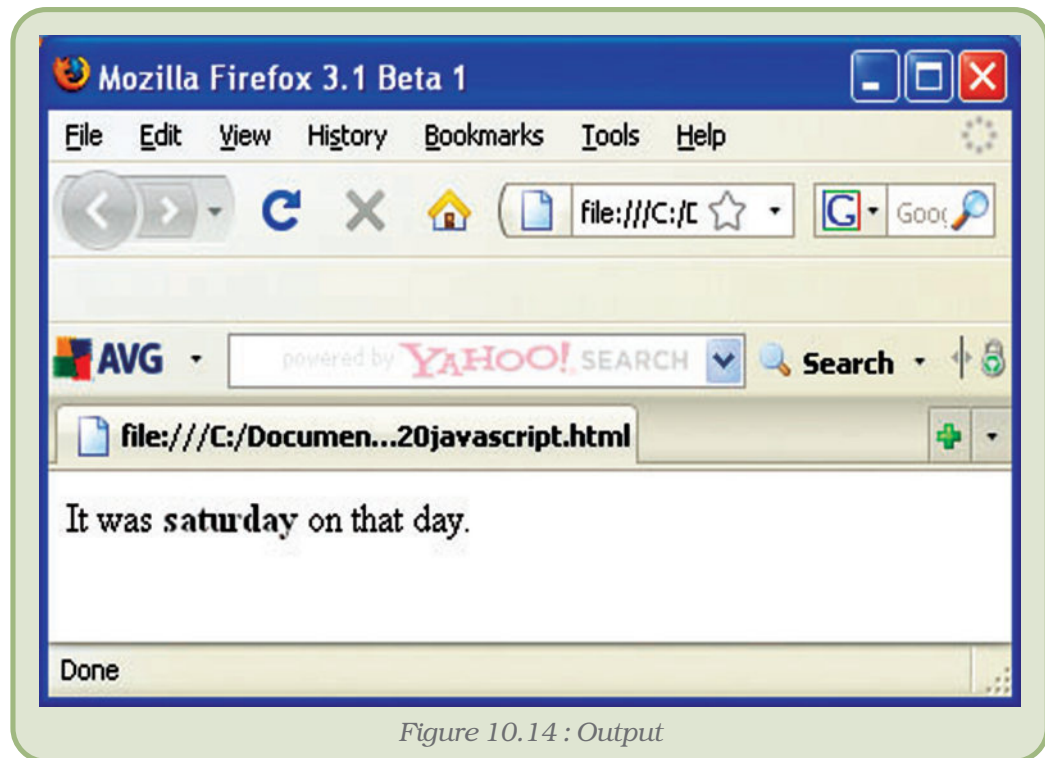
Figure 10.14 : Output

condition is met. JavaScript supports following looping statements :

- for
- do ... while
- while loop

Most loops have a counter variable which is initialised before the loop starts and then it is tested as part of the condition (expression) evaluated before every iteration of the loop. Finally, the counter variable is incremented or updated at the end of the loop body just before the condition is evaluated again.

For

The **for** loop consists of three optional expressions separated by semicolon, followed by a block of statements executed in the loop. Loop statements executed repeatedly again and again until the condition is false. The **for** loop is used when we know in advance how many times the script code should run.

The Syntax is

for([*initial-expression*]; [*condition*]; [*increment-expression*])

{

statements

}

Parameters

Initial-expression   – used to initialise a counter variable.

Condition   – If condition evaluates to true, the statements are executed.

Incr.-expression   – to increment the counter variable.

Examples

The following **for** statement declares variable i and initialising it to 1. It checks that i is less than 20, performs the two succeeding statements, and increments i by 2 after each pass through the loop.

```
// for loop to display odd numbers between 1 to 20
for (var i = 1; i < 20; i+=2)
{
    document.write(i);
    document.write("<BR>");
}
```

Program 10.6 : A JavaScript program to generate the table of number 7 using **for loop** statement.

```
<HTML>
<HEAD>
    <TITLE> Table of 7 </title>
</HEAD>
<BODY>
<SCRIPT language="JavaScript" type="text/JavaScript">
    document.write("<H2> Table of number 7 </H2>");
    for(i = 1; i <= 10; i++ )
    {    document.write(7*i);
         document.write("<BR>");
    }
</SCRIPT>
</BODY>
</HTML>
```
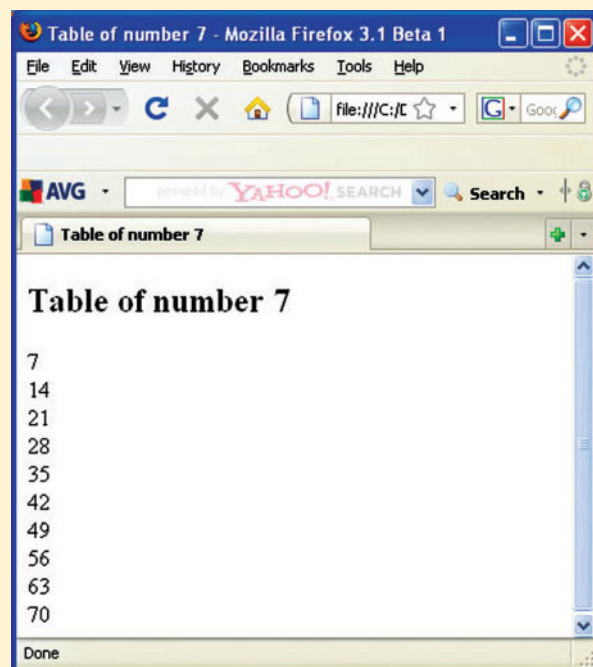
While

The *while* loop statement is simpler than the *for* loop. It consists of a condition and block statement. The condition is evaluated before each pass through the loop. If the condition is true then it executes block statement.

The Syntax is

> while (*condition*)
>
> {
>
> statements
>
> }

Example : The following while loop gives same output as for loop in previous example.

```
// While loop to display Odd numbers between 1 to 20
var i = 1;      // Initialization of counter variable
while (i < 20)  // Condition
{
    document.write(i);
    document.write("<BR>");
    i++;        // Updation
}
```

In while loop, we have to maintain a counter variable which controls the execution of statements in the loop.

Program 10.7 : Write a JavaScript program to generate first 10 Fibonacci numbers.

```
<HTML>
<HEAD>
<TITLE>Fibonacci Series</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/JavaScript">
// Program to print the Fibonacci series upto 10 numbers
document.write("Fibonacci Series... upto 10 numbers <BR>".fontsize(4));
//.fontsize to increase the font size of the string
i=0;
document.write(i + "  ");
j=1;
document.writeln(j + "  ");
var x = 3;
while (x <= 10)
{
    t = i + j;
    document.write(t + "  ");
    i = j;
    j = t;
```

```
    x++;
}
</SCRIPT>
</BODY>
</HTML>
```

Fibonacci Series - Mozilla Firefox

File   Edit   View   History   Bookmarks   Tools   Help

file:///C:/Documents%20anc   Google

Customize Links   Free Hotmail   My Yahoo!   Windows Marketplace   Windows Media   »

Fibonacci Series... upto 10 numbers
0 1 1 2 3 5 8 13 21 34

Done

Do...While

The do...while loop is much like a while loop. It will repeat the loop until the specified condition is false. This loop will always be executed at least once, even if the condition is false, because the block of statements is executed before the condition is tested. In this loop statement, curly braces are optional.

The Syntax is

> do
>
> {
>
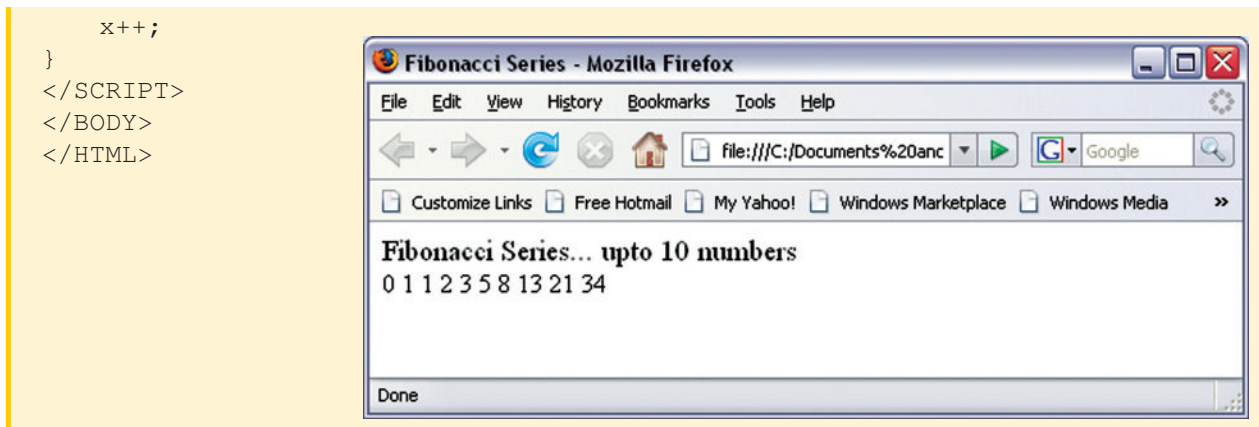> *statements*
>
> }
>
> while (*condition*);

Example : The following do...while loop gives same output as while loop in previous example.

```
// do…while loop to display Odd numbers between 1 to 20
var i = 1;  // Initialization of counter variable
do
{
    document.write(i);
    document.write("<BR>");
    i++;  // Updation
}
while (i < 20); // Condition
```

Condition lies between the parentheses after the block of statements with while keyword.

10.10.3 LABEL

A label is an identifier followed by a colon that can be helpful in directing the flow of program.

The Syntax is

label: *statement*

The value of label may be any JavaScript identifier. The statement that you identify with a label may be any statement.

Example

In this example, the label "whileloop" identifies a while loop.

```
x=1;
whileloop:              // Label
while(x<=10)
{
document.write(x);
x++;
}
```

10.10.4 BREAK

Break statement is used to exit from the innermost loop, switch statement, or from the statement named by label. It terminates the current loop and transfers control to the statement following the terminated loop.

The Syntax is

break [*label*]

The break statement includes an optional label that allows the control to exit out of a labeled statement.

Example : The following program segment has a break statement that terminates the while loop when it is equal to 3.

```
var i = 0;

while (i < 6)

{

    if (i == 3)

break;              //the control moves out of loop in first iteration

i++;

}

document.write(i);
```

10.10.5 CONTINUE

The continue statement skips the statement following it and executes the loop with next iteration. It is used along with an *if* statement inside while, do-while, for, or label statements.

The Syntax is

continue [*label*]

   The continue statement does not terminate the loop. Instead, in a **while** loop, it jumps back to the condition and in a **for** loop, it jumps to the update expression. The **continue** statement can include an optional label that allows the program to terminate a labeled statement and continue to the specified labeled statement.

Example

A program to input 50 elements using prompt() and then compute sum of marks more than 40 using continue statement.

```
var marks = new Array();
var i = 0, sum=0;
while (i < 50)
{
    i++;
    // parseInt converts string value into a number.
    marks[i]=parseInt(prompt("Enter marks"));
    if (marks[i] <= 40)      // when the condition is true then
    continue;   // control goes to while condition expression.
    sum = sum + marks[i];
}
document.write(sum+"\n");
```

10.11 OBJECT HANDLING STATEMENTS

JavaScript provides some special statements to handle the objects. Two type of statements are **for...in** and **with**

10.11.1 FOR...IN

The **for...in** statement iterates a specified variable over all the properties of an object.

The Syntax is

   for (<*variable*> in <*object*>)

   {

   statements

   }

   The body of the for...in statement is executed once for each property of an object. Before the loop statement is executed, the name of one of the object's property is assigned to variable, as a string. We can use this variable to look up the value of the object's property with the [ ] operator.

Example : To print the name and value of each property of a Book object.

309

```
<HTML>
<HEAD>
<TITLE> for... in Example </TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript" TYPE="text/javascript">
var Book = new Object();          // Object creation
// Properties and values of Book
Book =    {      Title:"The Discovery of India",↵
          Author:"Jawahar Lal Nehru", ↵
          Publisher: "Penguin Books", ↵
          Price:Rs 399//- ↵
     };
var result = "";
// Name of distinct property of Book assign to b in each loop
execution
for (var b in Book)
{
    // Book[b] is used to get the values.
    result += "Book." +  b + " = " + Book[b] + "<br>";
}
// To print names and values of each property of Book object.
document.write(result);
</SCRIPT>
</BODY>
</HTML>
```
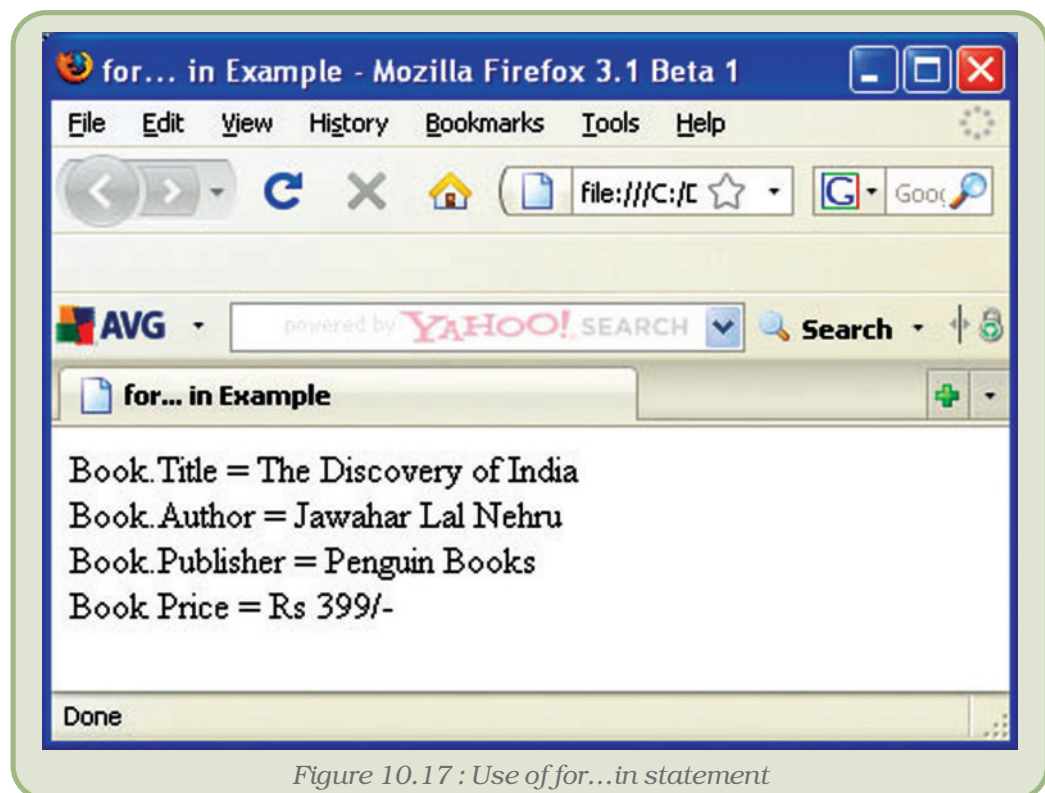
Output

Figure 10.17 : Use of for...in statement

10.11.2 WITH

With statement saves a lot of typing when properties of same object have to be accessed. For example, it is common to work with deeply nested object hierarchies. Sometimes we have to type expressions like this one to access elements of a HTML form :

Frames[1].document.forms[0].fname.value

Examples

The following script illustrates the use of **with** statement. Here two object Math and document are default objects. The statements following the **with** statement along with *Math* object refer to the PI property and the cos and sin methods, without specifying an object. Same way the statement following the **with** statement along with *document* object refer to the write method, without specifying the object. JavaScript assumes the Math and document object for these references.

```
var area, circumference
var r=10
with (Math)
{
    area = PI * r * r
    circumference  = 2*PI*r
}
with (document)
{
    write("Area of the Circle: "+area+"<br>");
    write("Circumference of the Circle: "+circumference);
}
```
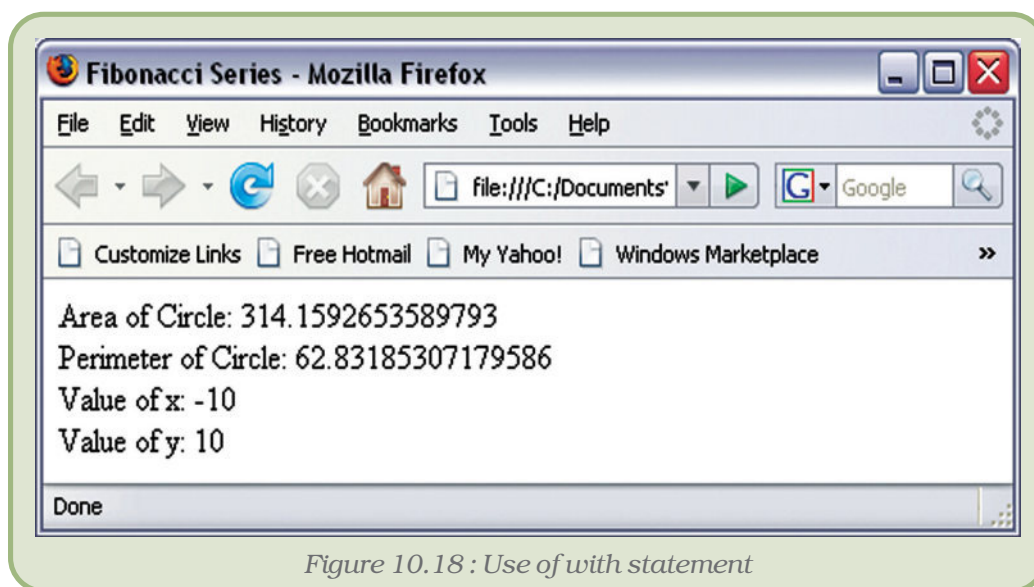
Output

Figure 10.18 : Use of with statement

311

10.12 JAVASCRIPT FUNCTIONS

Function is a named block of statements which can be executed again and again simply by writing its name and can return some value. It is useful in making a program modular and understandable.

10.12.1 DEFINING A FUNCTION

Function can be defined using the following syntax:

The Syntax is

```
function <function-name>([<parameter list>])

{

    ... body of the function ..

}
```

The function definition begins with keyword function, followed by function's name and optional parameter-list within parenthesis.. Braces are used ({ and }) to enclose all of the statement in a function. Let's an example of function definition.

```
function Welcome()
{
    alert("Welcome to NCERT ");
}
```

These statements define a function Welcome that displays an alert message to the user.

10.12.2 USING PARAMETERS WITH FUNCTION

The arguments received by the function in a list of corresponding values are called parameters. These values can be assigned to local variables within the function. Let's try an example of function using parameter :

```
function Welcome(name)
{
    alert("Welcome to NCERT , " + name);
}
```

We have learnt how to create simple functions. The best place for a function definition is within the <HEAD> section of the HTML document, because the statements in this section are executed first, this ensures that function is defined before it is used.

Example : Using a function within the HEAD section of a HTML document.

```
<HTML>
<HEAD>
<TITLE> Define a Function </title>
```

```
<script language="JavaScript" type="text/JavaScript">
function Welcome(name)
{
    alert("Welcome to NCERT, " + name+"!");
}
</script>
</HEAD>
<BODY>
This is the body of a HTML document. You will observe that I
have used script code in small case and other HTML tags in
Capital to enhance the readability of JavaScript Code.
</BODY>
</HTML>
```

10.12.3 CALLING THE FUNCTION

A function can be called by writing the name of the function along with the list of arguments. A function call can also be used in an event handler code also.

The Syntax is

<function name> ([<parameter list>])

Example

```
Welcome("MANYA");
```

This tells the JavaScript interpreter to transfer control to the first statement of function 'Welcome'. It also passes the parameter "MANYA" to the function. The value will be assigned to the ***name*** variable inside the function.

Program 10.8 : A HTML document that illustrates function definition and calling of Welcome() function.

```
<HTML>
<HEAD>
<TITLE> Defining and calling a Function </title>
<script language="JavaScript" type="text/JavaScript">
// function definition
function Welcome(name)
{
    alert("Welcome, " + name);
}
</script>
</HEAD>
<BODY>
<H1> A Function Example</H1>
This example illustrates use of function and popup boxes.
<br>
<script language="JavaScript" type="text/JavaScript">
    var nm = prompt("What's your name? ", "Your name please…");
    // Calling of function Welcome().
```

```
    Welcome(nm);
</script>
</BODY>
</HTML>
```
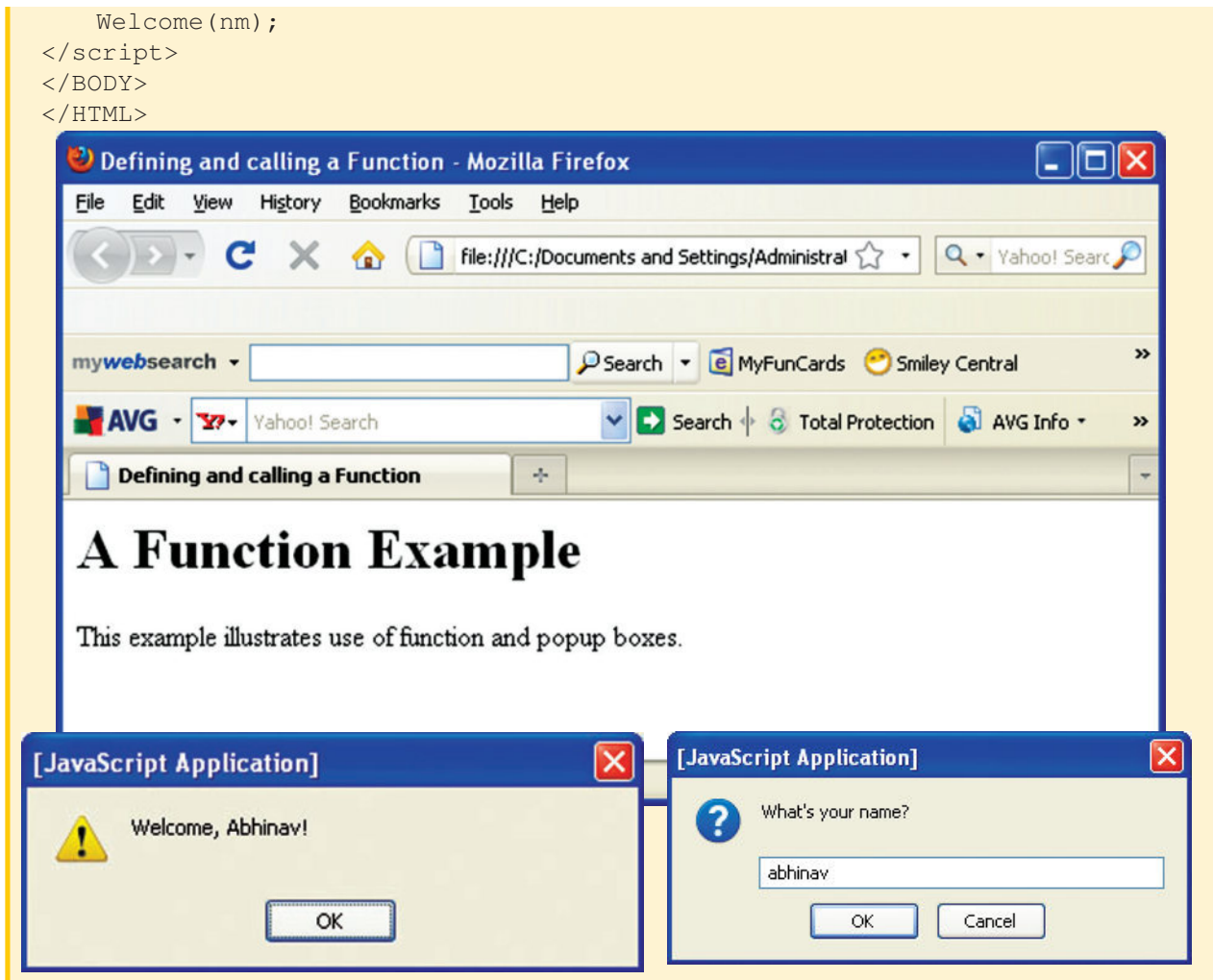

10.12.4 CALLING A FUNCTION FROM AN EVENT

Once a function is defined, it may be used with events like on Click event. Here, the function simply becomes another JavaScript command. For example:

```
<INPUT type = "button"
    value = "Calculate"
    onClick = calc() >
```

When the user clicks the button, the program automatically calls the calc() function.

10.12.5 RETURNING VALUE FROM FUNCTION

The *return* statement is used to return a value from a function. A variable using assignment operator can hold the returned value.

For example, a program to calculate simple interest using function is :

```
<HTML>
```

```
<HEAD>
<TITLE>A Simple JavaScript Function returning Value </TITLE>
<script language="JavaScript" type="text/JavaScript">
    function si( p, r, t )
    {    var s = (p * r * t)/ 100
         return s;   // function returning value s
    }
</script>
</HEAD>
<BODY>
<script language="JavaScript" type="text/JavaScript">
    var result = si( 1000, 5, 7)   // returned value assigned
to result
    document.write ( "Result = " + result);
</script>
</BODY>
</HTML>
```

Output

Figure 10.20 : Output

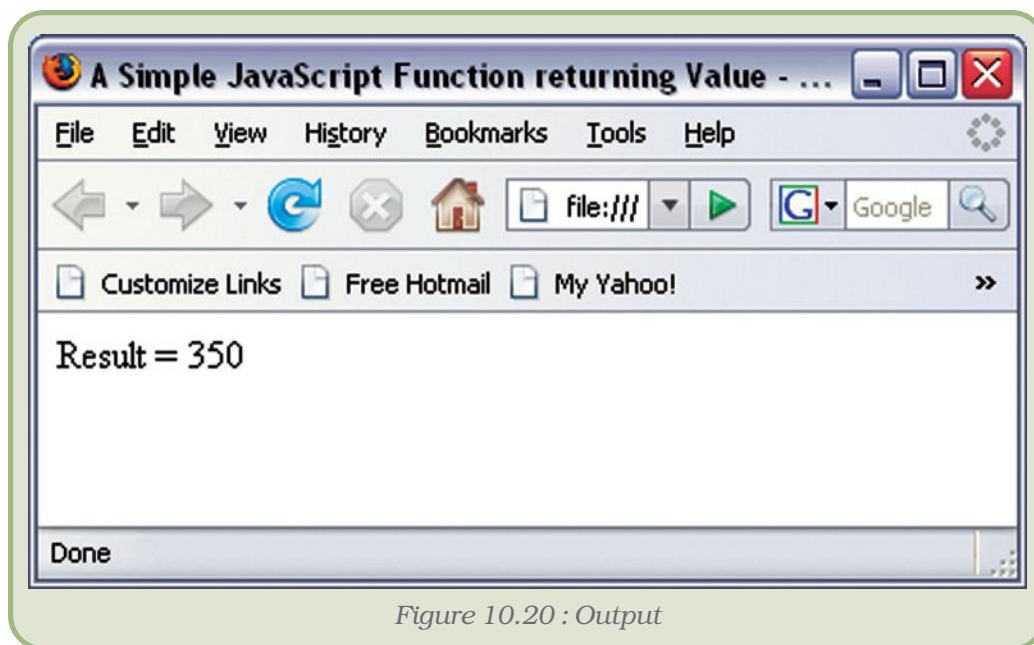## 10.12.6 NESTED FUNCTION

A function may be nested inside another function definition. They may not be defined within statement blocks, such as the body of an if statement or while loop etc.

Suppose we want to calculate the area of the circle, which is equal to $3.14 \times r^2$. The nested functions would be written as follows:

```
function Area(r)
{
    function Square(x)
    {
```

315

```
            return x*x;
        }
        return 3.14*Square(r);
    }
```

In this script, Square() function is nested inside the Area() function.

RESERVED/KEY WORDS IN JAVASCRIPT

abstract	boolean	break	byte
case	catch	char	class
const	continue	debugger	default
delete	do	double	else
enum	export	extends	false
final	finally	float	for
function	goto	if	implements
import	in	instanceof	int
interface	long	native	new
null	package	private	protected
public	return	short	static
super	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	

Summary

- JavaScript is a platform independent object-based scripting language.
- Client-side JavaScript is the name given to JavaScript code that is executed by a web browser on client machine.
- JavaScript is a case-sensitive language and all the statements are written in lower case.
- JavaScript allows omitting semicolon when statements are placed in separate lines. If we combine statements into a single line, we must use semicolon to separate them.
- ***document.write*** is a standard JavaScript command for producing output to a document window.
- Literals refer to the constant values that are used directly in a program code.
- A variable is a container of values or string. The values stored in a variable can be accessed using the variable name.
- JavaScript supports three primitive data types: number, string and boolean. JavaScript allows two composite data types: objects and arrays.
- Expression is combination of operators and literals or variable names.

316

- The increment and decrement operators act on only one operand. These are used for increasing or decreasing the value of a variable by 1.

- An assignment is a basic operator, which assigns the value of its right operand to its left operand. This operator is represented by an equal sign(=).

- Relational operators are used to compare the values of operands and it returns Boolean value based on the condition.

- JavaScript has three logical operators: **&&** (AND) ,   **||** (OR) , and **!** - (NOT)

- The + operator is also used to concatenate two string operands. It gives priority to string operands over numeric operands.

- Conditional operator (? :) is also called ternary operator.

- **new** operator is used to create an instance of a user-defined object type or one of the predefined object types.

- The **delete** operator is used to de-allocate the memory space.

- The *in* operator returns true, if the specified property/index exists in the specified object.

- Alert box is a dialog box with a text message and "OK" button.

- Confirm box is a box meant to verify or accept some information, the user have to click either "OK" or "Cancel" buttons to proceed.

- Prompt box allows us to gather user's input with the help of text field.

- Block statement combines two or more statements into a one statement. Block statements are commonly used with conditional and looping statements.

- An **if** statement is used to execute a statement or a block of statements on the basis of condition.

- A switch statement in JavaScript is used to perform different actions based on different conditions. It can be a replacement for multiple if... else if... statement.

- Loop statements tell JavaScript interpreter to execute same statements again and again until a specified condition is met.

- The **for** loop consists of three optional expressions. It executes block statement repeatedly again and again until the condition is false.

- The **while loop statement** is simpler than **for** loop. It repeats block statement again and again until the specified condition is false.

- Unlike **while** loop, **do...while** loop always executes a block of code at least once.

- Instead of exiting a loop, continue statement skips the statement following it and restarts a loop in a new iteration.

- The **for...in** statement iterates a specified variable over all the properties of an object.

- The **with** statement establishes the default object for a set of statements.

- Function is a named unit for the group of JavaScript statements. If a user needs to send values to a function, the values are enclosed in the parentheses after the function name and sent as a comma-separated list of arguments when function is called.

- The **return** statement is used to return a value from a function. A variable using assignment operator can hold the returned value.

- A function may be nested inside another function definition.

EXERCISES

SHORT ANSWER TYPE QUESTIONS

1. Is it possible to write programs for standalone applications in JavaScript?

2. Where should we use semicolon in the statements? Is it mandatory?

3. Differentiate between client-side and server-side JavaScript?

4. What is the purpose of **document.write()** in JavaScript code?

5. What are the invalid variable names in following? Explain with reasons.
   My_Name, number10, $100, father's name, marks%

6. What are reserved words? Can we use reserved words as identifiers?

7. What do you mean by literals?

8. What are the data types in JavaScript?

9. What is the purpose of **var** statement?

10. Why strings are enclosed within the single or double quotes?

11. Write JavaScript code that displays the following:

    He said, "Mahatma Gandhi was a non-violent soldier of India."

12. Can we use different data type values in a single array?

13. Write a statement to assign numeric values 65, 75, 80, 87, 90 to an array named **stumarks**.

14. Are the null and zero have similar values?

15. What are operators? What is the significance of an operator?

16. What type of a value a *prompt()method returns*?

17. Write the equivalent JavaScript statements for the following:
    (a) Area = $3.14r^2$
    (b) KE = $\frac{1}{2} mv^2$

18. What will be the value of variable **r**?
    var r = 45 % 7;

19. What will be the value of variable r?
    var a = 4;
    var b = 7;
    var c = "NCERT";
    r = a + b + c;

20. The '==' operator is not the same as the assignment operator '='. (True or False).

21. Find the value of variable *total*?
    total = (360 * 5) + ((40 / 8) - 9) - ((14 * 6) / 2);

22. What is the value of variable b?
    a = 25;
    b = (++a) + 7;

LONG ANSWER TYPE QUESTIONS

1. Write a JavaScript program to find out whether the given year is leap or not. Use prompt box for input the year.

2. Write a JavaScript program which welcomes the user by addressing
   Good Morning, <User_name> during hour in time is 4 to 11
   Good Noon, <User_name> when hour value is 12.
   Good Afternoon, <User_name> when hour value is 13 to 16
   Good Evening, <User_name> when hour value is 17 to 23
   Good Midnight, <User_name> when hour value is 24 or 1 to 3.

3. Write a program to find the greatest number among three given numbers?

4. What happens when the following JavaScript code is executed?
   var get_res = confirm("Did you like this chapter?");
   if (get_res == true)
   alert("Okay! Let's go to the next!");

5. What will be the output if variable **marks** is 75?
   if (marks >= 80)
       {
   alert("Excellent!");
       }
   else if (marks >= 60 && marks < 80)
       {
   alert("Good!");
       }
   else if (marks > 50 && marks < 60)
       {
   alert("Average!");
       }
   else
       {
   alert("Improve yourself !");
       }

6. Write a program using switch statement to print word equivalent of a number from 0 to 9, e.g. 3 should be displayed as "Three".

7. Write a statement that displays an alert box which looks like this :

8. Write a program to print the following output using *for* loops.
   1
   22
   333
   4444

9. How many times will the following *for* loop be executed?
   for (a = 0; a <= 10; a = a + 2)
   {
   ... statements ...
   }

10. What will be the final value of the variable **sum**?
    var sum = 0;
    function add()
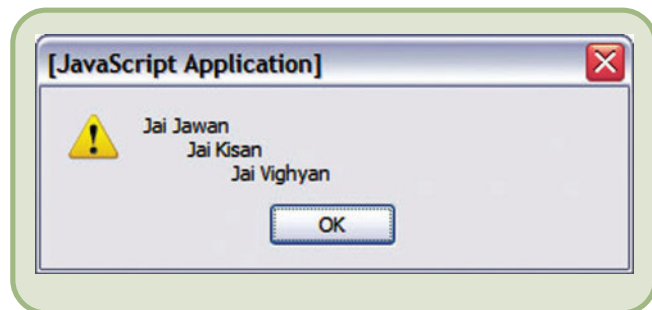
```
{
var sum = sum + 20;
}
add();
sum = sum + 1;
```

11. What will be displayed in the alert box at the end of script execution?
```
var y = 0;
for (x = 0; x <= 5; x++, y = y + 50)
{
y = y + 10;
}
alert("The value of y is :" + y);
```

12. Write a program to calculate the average of 5 numbers entered by the visitor through a prompt?

13. If the value of variable **num** is 30, how many times will the following *while* loop be executed?
```
while (num <= 30)
{
... statements ...
num = num + 3;
}
```

14. Write a program to find the reverse of a number (i.e. reverse of 123 is 321)

15. Write a program to convert a decimal number into a binary number.

16. Write a program to check whether a number is palindrome or not. A number is palindrome if it is equal to its reverse number.

17. Using continue display the odd numbers between 1 to 20.

18. Write a program to find sum of digits of a number, e.g. 453 results 4+5+3=12.

19. Write a program to generate prime numbers up to a specific limit.

20. What will be the output of the following JavaScript code?
```
for(i=1; i<=5; i++)
{
document.write("<BR>")
for(j=1; j<=i; j++)
document.write("*")
}
```

21. Identify the errors in the following code segment:
```
function 3_alert_box
{
alert("Hi!, I am from a function.');
}
```

22. Write a program using *while* statement to find out the sum of first n numbers.

23. Identify the errors in the following code segment:
```
fun_alert_box
    {
```

```
document.write("Hi!, I am from a function);+1
    }
```

24. What do you understand by nested function?

MULTIPLE CHOICE QUESTIONS

1. The file extension for external JavaScript file is —
   (i) .jscript
   (ii) .js
   (iii) .jav
   (iv) .java
2. Why comments are used within JavaScript programs?
   (i) To tell the browser there is HTML in our JavaScript code.
   (ii) Because HTML does not have its own command for comments.
   (iii) To explain what a script does.
   (iv) None of the above.

3. The original name of JavaScript was —
   (i) JavaScript
   (ii) LiveScript
   (iii) WireScript
   (iv) ECMAScript

4. Which of following is not a valid expression?
   (i) 7.5
   (ii) b + a = c
   (iii) {feet:10, inches:5}
   (iv) [2,3,6,9,5,7]

5. What will be the value of **res** in the following expression?
   var res = "70" + 25
   (i) 95
   (ii) 7025
   (iii) 25
   (iv) No output due to error.

6. What will be the value returned by expression "India" < "bharat"?
   (i) 0
   (ii) true
   (iii) false
   (iv) Invalid expression

7. **delete** operator is used to delete the
   (i) .js files.
   (ii) cookies.
   (iii) created objects.
   (iv) functions.

8. To which object does the confirm() method belongs to?
   (i) document
   (ii) window
   (iii) frame
   (iv) date

9. Which of the following statement is used to repeat execution of block of the statements?
   (i) if...else...
   (ii) continue

321

(iii) while
(iv) switch

10. What will be value of sum after execution of the statement?
for(i=1, sum=0; i<=5; i++) sum+=i;
 (i) 13
 (ii) 15
(iii) 25
(iv) Error in statement.

11. Which of the loop executes a block of code at least once?
 (i) while
 (ii) for
(iii) do...while
(iv) for/in

12. Which of the following loop statement is used to iterate a specified variable over all the properties of an object?
 (i) for
 (ii) do...while
(iii) for/in
(iv) none of these

Activities

- Write an HTML document using JavaScript code, which displays your name, class and name of the school.

- Write an HTML document using JavaScript to change background's colour in your web page randomly. (Hint: use document.bgcolor property and Math.random() function.)

- Write a program to check whether the user is eligible to cast the vote. The user must check following two conditions for casting the vote:
 (a) the age should be equal to or greater than 18 years and
 (b) the nationality should be Indian.

- Write a simple Quiz program that asks the user 5 questions. Alert about answers of the questions and give the user a score at the end of the Quiz.

- Write an HTML document using JavaScript code to validate the form in your website.

References

JavaScript – The definitive Guide - David Flanagan, O'REILLY

SAMS Teach Yourself JavaScript in 24 hours – Michael Moncur, Sams Publishing

JavaScript – A Beginner's Guide – John Pollock, McGraw-Hill.

http://www.webreference.com/programming/javascript

http:// www.webdevelopersnotes.com/tutorials/javascript

http://www.javascriptkit.com

http://www.w3schools.com/js

http://sun.com

http://www.mozilla.org

PROJECT BASED LEARNING

11

Objectives

After completing this Chapter, the student will be able to:

- *Know about project and project based learning*
  - ➢ *explore the possibilities of executing a project,*
  - ➢ *experience the concept of 'learning by doing',*
  - ➢ *seek solutions to problems, using CCT tools.*

- *Understand the various processes involved in project based learning and the importance of team work in project based learning*
  - ➢ *develop projects for various real life situations,*
  - ➢ *work in realistic contextualised problem-solving environments,*
  - ➢ *realise the success of a project by experiencing the desired output.*

- *Apply knowledge and understanding of project based learning processes in new situations*
  - ➢ *improve communication skills,*
  - ➢ *enhance self-confidence,*
  - ➢ *appreciate that a project can be executed in different ways,*
  - ➢ *build up teamwork and leadership skills.*

"An idea that is developed and put into action is more important than idea that exists only as an idea."

Gautam Buddha

Introduction

Project based learning gives a thorough practical exposure to a problem upon which the project is based. Projects are developed generally in groups where students can learn various things such as working together, problem solving, decision making, and investigating activities. Project based learning involves the steps such as analysing the problem, formulating the problem into small modules, applying the mechanism or method to solve each module and then integrating the solution of all the modules to arrive at the complete solution of the problem. To solve a problem it is required that those who work on it gather the relevant data and process it by applying a particular method. Data may be collected as per the requirement of the project in a particular format. All the team members should be associated to accomplish the task. After collecting data, it should be processed to solve the problem. The results should be reported in a predetermined format.

11.1 MODULAR APPROACH TO PROJECTS

There are several approaches to execute a project such as modular approach, top down approach and bottom up approach. Amongst the most common forms of project based learning, the modular approach to execute a project is the most reliable technique. A structured or a modular approach to a project means that a project is divided into various manageable modules and each of the modules has a well defined task to be performed with a set of inputs. This would lead to a set of outputs which when integrated leads to the desired outcome.

Steps in Project Based Learning

The different steps involved in Project Based Learning figure 11.1 are :

1. **Identification of a project:** The project idea may come through any real life situation. For example one could think of doing a project for organising a seminar. One needs to understand the usefulness of the project and its impact. Students must be encouraged to undertake interdisciplinary projects.

2. **Defining a plan:** Normally for any kind of project, there are several project members involved in it. One project leader has to be identified. The roles of project leader and each project member have to be clearly defined. Students who are performing a project must be assigned with specific activities. The various tools for executing these activities must be known. To obtain a better solution, one should always think of the extreme situations.

3. **Fixing of a time frame and processing:** Every project is a time relevance project. A student must understand the importance of time frame for completion of the project. All the activities which are

Figure 11.1 : Steps in Project Based Learning
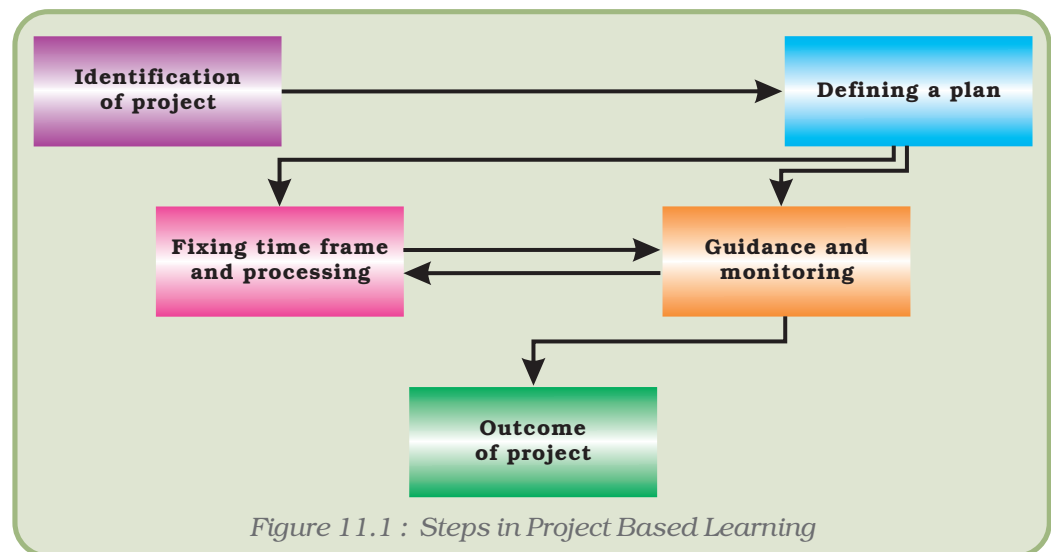
performed in the projects require certain amount of time. Every project must be well structured and at the same time it must be flexible in its time frame.

4. **Providing guidance and monitoring a project:** Many times, the participants in the project get stuck up with the process and become unable to proceed further. In such a case or otherwise they need guidance. Guidance can be obtained from various resources such as books, websites and experts in the field. While it is essential that project leader should ensure monitoring of the project, the guide teacher also helps in monitoring the project.

5. **Outcome of a project:** One needs to understand what can be the outcome of the project. The outcome can be single or it can be multiple. The outcome of the project can be peer reviewed and it can be modified as per the feedback from users or experts.

Let us consider a project "Organising a seminar for career counselling in a school". The project can be broken into sub–tasks such as :

- Drafting the concept note that would establish the need for organising such a seminar,
- Arranging a hall for the seminar having proper sitting arrangements with audio–visual facilities,
- Making a list of participants and sending information to the participants,
- Sending invitations to the guests,
- Arranging career counsellors to speak at the seminar,
- Arranging for hospitality to the guests, etc.

The above example demonstrates the need of a modular approach to a project. A separate team of people may be assigned to perform each of the above listed sub–tasks.

The advantage of modular approach is that it is much easier to design a smaller task. Also, some of the modules can be re–used if they have already been done for some other project. For example the list of participants may be re–used if it was already prepared for some other seminar. It also ensures that some tasks can be done in parallel which is very helpful for executing a project in short time.

11.2 TEAMWORK

Many real life tasks are very complex and require a lot of people to contribute in achieving them. An effort made by many people together in order to accomplish a task is called **teamwork**.

For example, in many sports, there is a team of players. These players play together to win a match. Take an example of a cricket

team. We find that even if a bowler bowls a good ball but if the fielder cannot take a catch then a wicket cannot be taken. So, in order to take a catch we require efforts of a bowler and also of fielders. To win a cricket match, contributions from all the team members in all the three areas batting, bowling and fielding are required.

11.2.1 COMPONENTS OF TEAMWORK

Apart from technical proficiency, a wide variety of other components make a successful teamwork. It comprises of skilled team members with specific roles, to achieve the goal.

Communicate with Others

When a group of people performs one job, it is necessary to have an effective communication between the members of the team. Such communication can be done via e-mails, telephones or by arranging group meetings. This helps the team members to understand each other and to sort out problems to effectively achieve the goal.

Listen to Others

It is necessary to understand the ideas of others while executing a job together. This can be achieved when the team members listen to each other in group meetings and follow steps that are agreed upon.

Share with Others

Ideas, images and tools need to be shared with each other in order to perform a job. Sharing is an important component of teamwork. A person in the team who is well versed in a certain area should share his expertise and experience with others to effectively achieve the goal within the time frame.

Respect for Others

Every member of the team must be treated respectfully. All the thoughts and ideas that are put forth in the group meetings may be respected and duly considered. Not respecting the views of a particular member may cause problems and that particular team member may not give his best.

Help Others

A helping hand from every member is a key to success. Sometimes help from people who are not a part of the team is also obtained in order to accomplish a job.

Participate

All the team members must be encouraged by each other to participate in completing the project and also in discussions in group meetings. Also every member should take an active participation so that they feel their importance in the team.

11.3 A SUCCESSFUL LEARNING EXPERIENCE

In order for projects to be successful learning experiences, each student should:
- have a sense of participation,
- have enough time to work on a project,
- have some choices in the creation of questions, approaches, etc.,
- possess the skills to work with others,
- have the knowledge necessary to explore questions that arise,
- understand the teacher's method of evaluation.

Some of the projects with the objective to provide exposure to students to undertake and implement various aspects of real life projects, such as creating a magazine, Awareness of Pulse Polio Programme, Robotics, Health Analyser System, farewell to class XII, Know your District/Region, create a website as teamwork and establishment of a computer network are given below:

11.3.1 SCHOOL MAGAZINE

Project Title : Creation of School Magazine

Description

The project aims at creating a monthly magazine of your school (Figure 11.2), which should include articles describing the latest happenings and achievements in the field of sports, academics and other various activities that have occurred during the past month. It should not include any article that is a copy of copyright material. It must be created by the students only.

Figure 11.2 : Creating school magazine

Guidelines

Make a group of students to undertake a project on Producing School Magazine. This group should be divided further into several subgroups of 2-3 students to carry out the work on similar lines as mentioned below:

(a) **Task force :** This group should be involved in motivating the students for writing and submitting the material for the school magazine.

(b) **Students Articles :** This group should collect the articles submitted by students on areas of their interest. Articles may also include quiz, humour, puzzles, and poems, etc.

(c) **Sports Section :** This project team should collect all the information regarding the sports activities of the school like any zonal or district level games, team members who took part, etc.

(d) **Achievements in the field of Academic/Cultural activities :** This team should collect materials on the cultural and academic events of the school.

(e) **Illustrations :** This group should create the illustrations, drawings, etc. for making the Magazine more attractive and informative.

(f) **Editing and layouts for printing :** Editorial group should confirm that the articles are original writings of the students. After verification, the editorial group may compile all the collaborative works done by the rest of the groups and should create a layout on the computer. After creating the layouts the editing groups should show their layout to the teacher under whose supervision the project is going on.

After the layout has been finalised the magazine should be printed and can be distributed among the students and teachers of the school.

Outcome

The School Magazine.

11.3.2 PULSE POLIO

Project Title : Report on "Pulse Polio Sunday" held at your village/town.

Description

The goal of the project is to help and create a monitoring report for the Pulse Polio Program in your region (Figure 11.3). The group should work to promote the Polio Eradication Movement taken up by the Government of India.
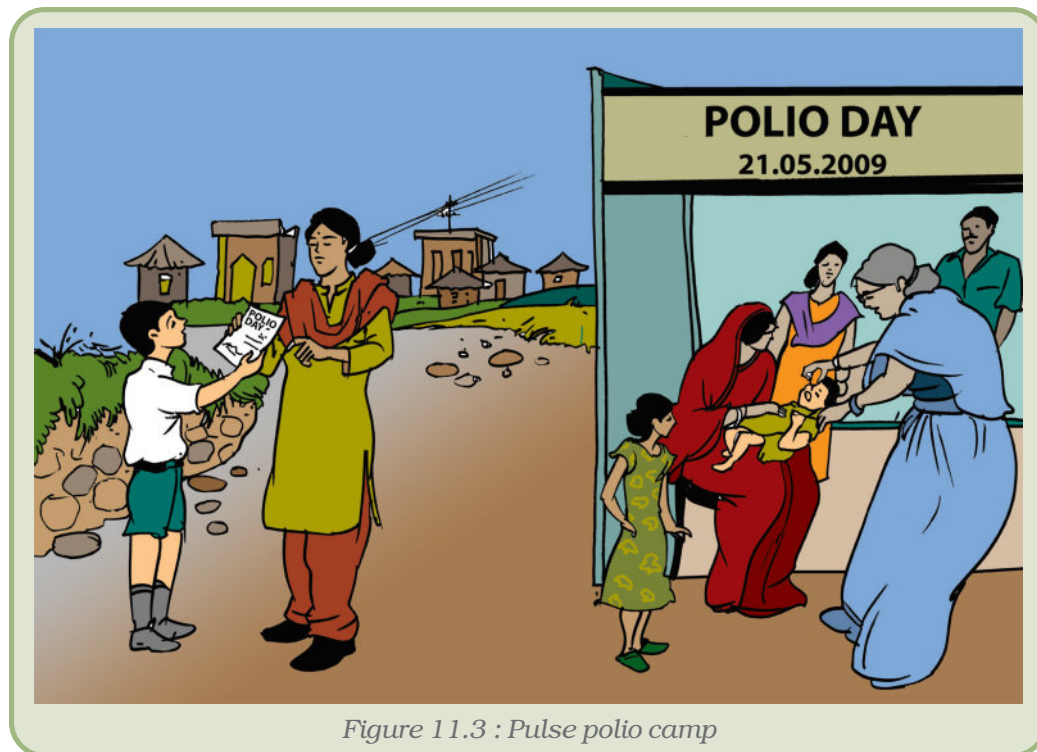
Figure 11.3 : Pulse polio camp

Guidelines

Let us create a group of 8–10 students. This group should be divided further into sub-groups of 2–3 students to carry out a specific work each covering different regions.

(a) **Task force :** The students in this group should contact the Pulse-Polio Programme coordinators at the pulse polio centre in their locality.

(b) **Motivation group :** This group should create digital posters on polio vaccination that can be used to promote pulse-polio awareness among people of their area.

(c) **Data collection :** After the Pulse Polio Programme scheduled for the month they should collect the data that how many children have been vaccinated and how many are still remaining in their locality.

(d) **Report generation :** The chart representing the data should be drawn using appropriate computer software.

The data of polio vaccination can be recorded for the whole year.

Outcome

"The Pulse Polio Report" highlighting the status of Polio Eradication Programme in your village/town created using some presentation software such as PowerPoint or Impress.

331

11.3.3 ROBOTICS

Project Title : Application of Robots in different fields

Description

The purpose of the project is to prepare a report on uses of robots figure 11.4 in different fields. It will also help students to realise the extent of automation and efficiency accomplished by robots in various fields of industry.
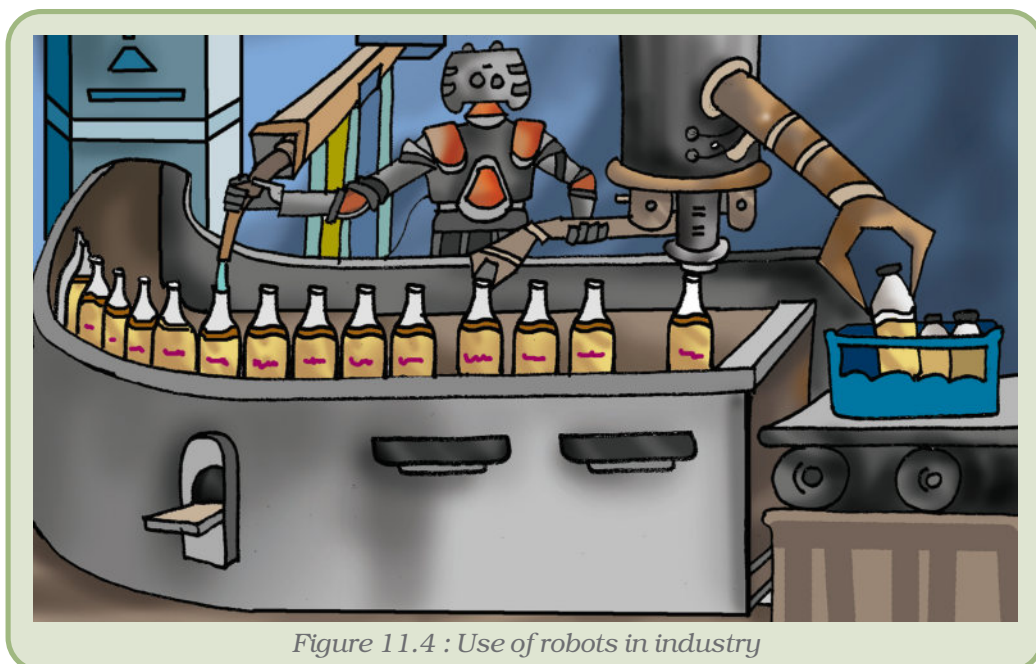

Figure 11.4 : Use of robots in industry

Guidelines

Form a group of 8–10 students for the project. This group should be divided further into subgroups of 2–3 students to carry out specific tasks on the similar lines as mentioned below :

(a) One subgroup should collect the information about different types of robots through websites or otherwise.
(b) Another subgroup should collect the information on various industries who are using specialised robots in their manufacturing units and to suggest the most suitable site for visit.
(c) Third subgroup should prepare a report on various functions/tasks performed by robots in the manufacturing unit.
(d) Fourth subgroup should prepare a module showing advantages of using robots.

Outcome

A compiled document describing the functions, and components used in robots and the tasks performed by them in a particular manufacturing workshop along with relevant information and pictures.

11.3.4 KNOW YOUR HEALTH

Project Title : Know your Health : Development of a System to record and analyse the data related to health of students in a class/school.

Description

This is a project related with collecting and analysing the data related to health of the students in a particular school/class. This project involves collecting data of students such as height, weight, their day to day food habits, their past health history etc. Some data related with their parents will also be collected (Figure 11.5).

The data will be collected by survey and this data will be stored in some Data Management System. It can then be analysed and some preliminary measures can be suggested to students and their parents.

This will help the students to maintain their health and hygiene and hence they can concentrate more on their studies.
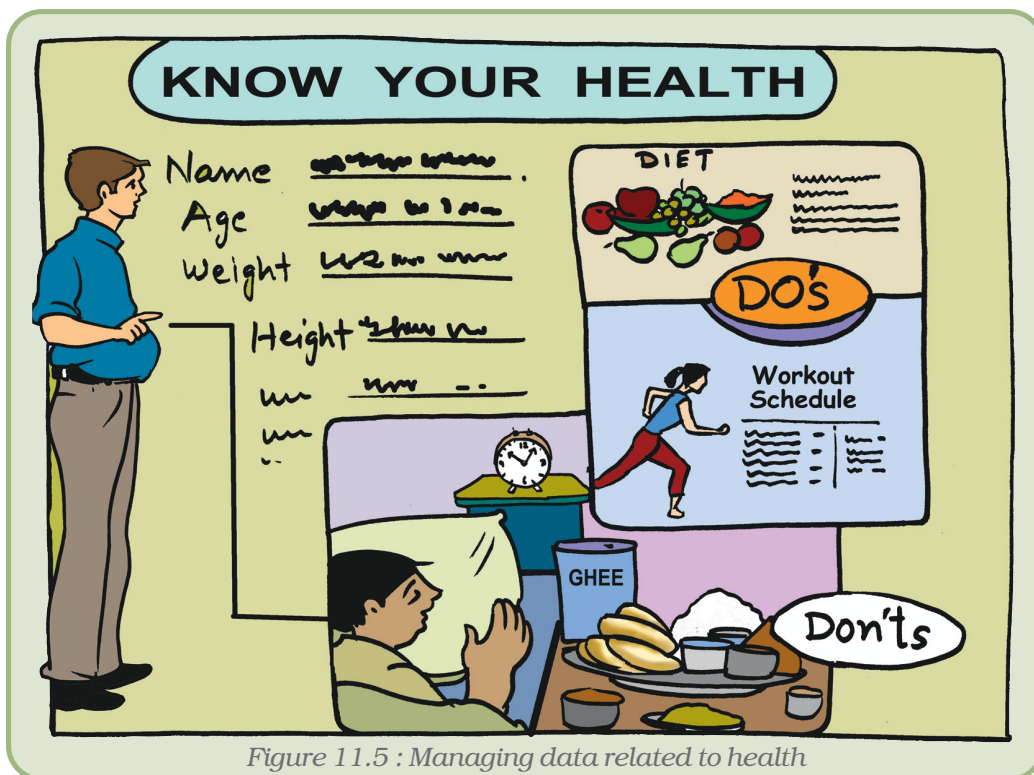

Figure 11.5 : Managing data related to health

Guidelines

(i) Decide a group of students from whom data is to be collected. A group of 2-3 students can collect data of 20 students.
(ii) Set the format of data to be collected by consulting a doctor. Note some useful information such as height, weight, food habits and exercise habits etc.

333

(iii) Collect the data by survey i.e. by distributing formats among students.

(iv) Get data entered into some database system such as MS Access/ MS Excel.

(v) Check the entered data.

(vi) Prepare some reports/graphs from collected data. Graphs can be student wise/class wise.

(vii) Show these graphs to doctor and obtain certain suggestions for each student.

(viii) Distribute suggestions to students/parents.

(ix) Take feedback from students/parents regarding use of this project and use it for improvement.

Format of data collection

1. Name of the student : ....
2. Class : ....
3. Age (in years) : ....
4. Height (in cm) : ....
5. Weight (in kg) : ....
6. Food Habits
   a. Number of times food taken (2/3/4) :
   b. Type of food (veg/non-veg)
   c. Regular food (Chapati, Rice, Dal, Veg)/Non regular food (fast food) :
7. Exercise habits
   a. Do you exercise daily?: (Yes/No) :
   b. If Yes, how many minutes?:
   c. Type of exercise: (Running/Play/on machine/Yoga/Surya Namaskar/other)
8. Time to wake up : ....
9. Time to go to bed : ....
10. Past medical history : ....
11. Write your normal daily schedule

Time	Activity
6 to 8 AM	
8 to 10 AM	
10 to 12 AM	
12 to 2 PM	
2 to 4 PM	
4 to 6 PM	
6 to 8 PM	
8 to 10 PM	
10 to 12 PM	

Outcome

Report on data related to health of students.

11.3.5 FAREWELL TO CLASS 12TH

Project Title : To organise a school event as 'Farewell to class 12th' as a collaborative project.

Figure 11.6 : A Farewell Party

Description

The aim of the project is to organise the event with collaboration of students and using CCT tools (Figure 11.6).

(a) **Identification of Participants**

The participants would be students of class 12th, class 11th, teachers, chief guest and special invitees. Among the participants, class 11th students are key members who will be organising the event. Class 12th students, teachers, chief guest and special invitees constitute the audience. The role of class 11th students, who are arranging the event, is more proactive than others. The first task is to get the list of students and teachers. One way of getting the names of the students of class 11th and 12th, is to ask one member of each section to get the details of the respective class. Another way is to get the information directly from class Attendance Register or school website.

(b) Planning

All events should start with a plan and there is a need for regular discussions among participants. Initially a group meeting is to be organised to discuss the objective, requirement and proposed line of action. Conventional way of holding a meeting is to inform students on the notice board or through class announcement. Another way of informing all the participants is by sending e-mails. An agenda of the farewell party is to be prepared and communicated to all members mentioning time, date and venue.

(c) Identification of Task

In the main event there are various subtasks which are to be delegated to different groups. Broadly the following tasks has to be assigned.

- Coordination of management for date, venue, chief guest and special invitees.
- Budget estimate.
- Collection of funds.
- Procurement of food and cold drinks.
- Procurement of gifts and mementoes.
- Organisation of cultural programme.
- Maintenance of accounts.
- Coordination and monitoring.

Guidelines

Use communication tools such as Messenger from any service provider such as Yahoo or own school e-mail service for creating task based groups. Members of each group should exchange information and get consolidated view. The decision can be then forwarded to coordination and monitoring group. The status of each task should be updated to the notice board or website or by sending the status report to all members through e-mails. This will create interest and enthusiasm in all the members whether he/she is involved directly or indirectly. A website can also be created where each member can post the status and notices regarding events.

(a) Delegation of Task

Once the subtasks are identified, the jobs have to be delegated to various groups. During allocation of tasks, it should be kept in the mind that representatives of all the classes are properly selected.

(b) Estimation

Conventionally the members allocate funds in different heads as per assumption and guess. Each group member can calculate the expected expense. For better estimation, members can do a market survey to explore the current market rate of various items and the prospective supplier. Market survey can be done directly or the same information

can be gathered from internet or by telephone. This will give fair idea about the expense and contribution from each student. The same can be informed to all the members with details of where and to whom the contribution is to be submitted.

(c) **Collection of Funds**

All the members can deposit their contribution to the designated member of their respective classes. The same can be updated in the database, and if required the same can be updated in the website also. A system can be developed to track the collection and expense.

(d) **Allocation of Funds**

As per the estimation each group would be allocated the funds. The group would regularly update the expenses, by keeping, bill/receipt of expenses made.

(e) **Cultural Programme**

There would be maximum participation of members in this group and would require coordination and regular meetings for making the event interesting. This task can be coordinated by e-mails. E-mail is a very effective tool for communication and can be used for communicating about rehearsal schedules. It will be better to use a unified messaging service for communication and event management.

Outcome

A good organisation of farewell party to outgoing students of class 12th.

11.3.6 KNOW YOUR DISTRICT

Project Title : Development of a system to record data related with your district to produce some useful information of the district.

Description

This is a project related with collection of information of the home district so that every student will be aware of the historical, geographical, political, cultural information of the home district. This will help the students to know many things about their home district. It ultimately helps them in their studies, since this information can be used in various subjects such as history, geography and science.

Guidelines

1. Make a group of students. This group should be divided further into sub-groups. Each sub-group is allotted one area of information to be collected such as one sub-group will collect historical information, second sub-group will collect the geographical information and third sub-group will collect the political information and so on.
2. Collect the information of the district as per the format given. It can be done by consulting the concerned teacher of the subject.

Such as group collecting historical data will consult with history teacher, group collecting geographical data will consult with geography teacher, and those collecting political data will consult with political teacher and so on.

3. Collect the data from various sources such as books, magazines, newspapers, Internet etc.
4. The concerned teachers should validate the collected data.
5. Enter the data in spreadsheet and prepare various charts and graphs so that the numerical information gets displayed in pictorial form.
6. Display these charts and graphs at various places in class/school.
7. Conduct small interactive sessions to make every student aware of the collected information.
8. Draw some conclusions from the collected information, which was normally not known, to the students.
9. Take the feedback from the students/teachers on the activity and use this for improvement.

Format for DATA COLLECTION

1. Historical information about the district
   (a) Date of formation:
   (b) Special features (tourist places/historical monuments/places of worship):
   (c) Earlier rulers:
   (d) History of the place (pre and post Independence) :

2. Geographical information
   (a) Population:
   (b) Area covered:
   (c) Number of villages:
   (d) Main productions of crops:
   (e) Number and names of rivers/lakes:

3. Educational information
   (a) Number of schools:
   (b) Number of colleges:
   (c) Number of students in school:
   (d) Number of students in college:
   (e) Name of universities:
    (f) Professional courses offered:
   (g) Number of Anganvadies:

4. Political information
   (a) Number and names of the assembly constituents:
   (b) Number and names of the parliament constituents:
   (c) Names of earlier MPs:
   (d) Name of present MP and MLA:
   (e) Political parties and their percentage of voting in a district:
    (f) Percentage of women participation in the local government body:

Outcome

A brief report on the history, geography and political information of your home district.

11.3.7 CREATING A WEBSITE AS TEAMWORK

Project Title : Develop a website to gather and fetch the information on the web.

Description

You are familiar with various websites on the Internet. You have also learnt various tools, and acquired skills needed to develop the website. Now you are able to create a website for school, company, hotel, club, family or personal use, etc. (Figure 11.7). The steps for developing website are given below :

Figure 11.7 : Creating a website

(a) **Planning :** The purpose of making the website should be very clear among all team members. Create a detailed document containing all the information about the website.

(b) **Designing :** After drafting the document, your next step will be to design all the web pages using any web editor.

(c) **Coding :** Now, the team is ready to validate web pages using Java Scripting language with proper documentation. The coding team needs to submit their code in time to the team leader. After

collection of all validated web pages, arrange them in proper location and folder.

(d) **Testing Website :** Once your website is ready, it will be tested by at least two members of the team for its smooth and performance based functionality. All members of the team must visit the website and also give suggestions to improve it.

(e) **Review :** After getting feedback from the team members, upload the website and submit it to any search engine, e.g. Google, Yahoo, bing etc.

Guidelines

1. Make it easy to verify the accuracy of the information on your site.
2. Ensure that there is a real organisation behind your site.
3. Ensure that honest and trustworthy people stand behind your site.
4. Make it easy to contact you.
5. Design your site so that it looks professional (or is appropriate for your purpose).
6. Avoid errors of all types, no matter how small they seem.
7. Make it user friendly i.e. easy to load, read, navigate, browse information and download etc.

Outcome

A complete website.

11.3.8 ESTABLISHMENT OF A COMPUTER NETWORK

Project Title : Development of a computer network in the school.

Description

Suppose a school has many computers but they are not connected with each other. The students want to connect these computers to each other and want to access the Internet. Computers can be connected through switch, hub, router and server. Switch gives the best performance compared to hub and router provides path to the entire network and server manages all the processes such as IP generation, restriction, security etc.

Students should collect the resources such as LAN cable, switch, router etc. After collection, they should connect LAN cable from computers to switch, switch to server, server to router and router to cable of the net. Now configure the router from server and give the IP addresses to all the computers of this network and check the connectivity of the server to client. If connectivity is there then computer network is ready to function.

Items Required

- LAN cards, switch, router and LAN cable with connecter.
- Two LAN cards for the Server computer.

Procedure

- Look at the figure 11.8 of Computer Network.
- First of all manage all the computers in the desired location.
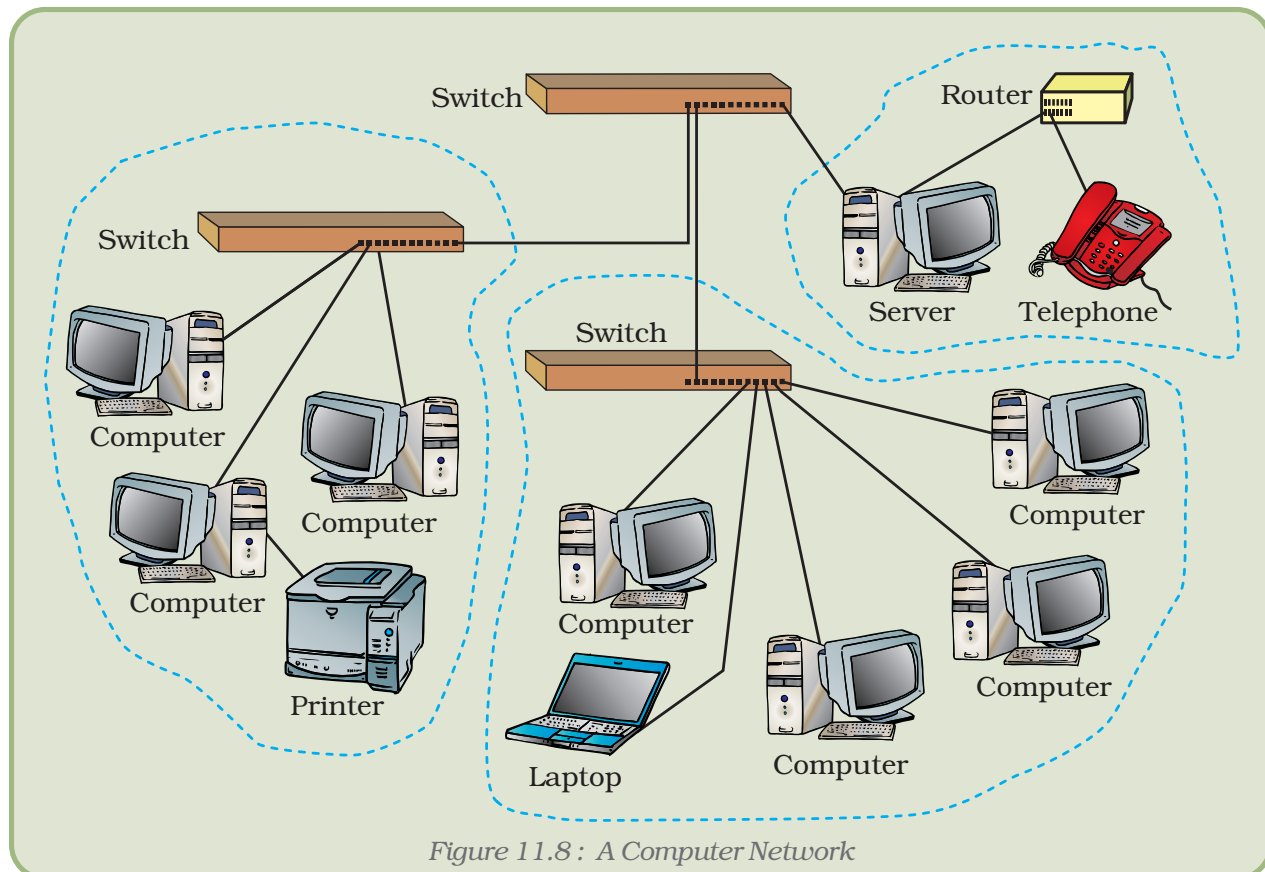- Cut LAN cable according to the distance of the computer and the switch.

Figure 11.8 : A Computer Network

- Connect computers and switch by the LAN cable.
- Connect the switch to the LAN card of server computer through LAN cable.
- Connect second LAN card of the server to the router.
- Connect the Router to the telephone line having connection such as ISDN or Broadband connection.
- After connecting all the computers to the switch and router define the class of the IP address.

Class A comprises network address 1.0.0.0 to 127.0.0.0; Class B contains network address 128.0.0.0 to 191.255.0.0; Class C network address's range from 192.0.0.0 to 223.255.255.0.

After defining IP range, configure Windows or Linux server.

Guidelines

- IP address of each computer should be unique.
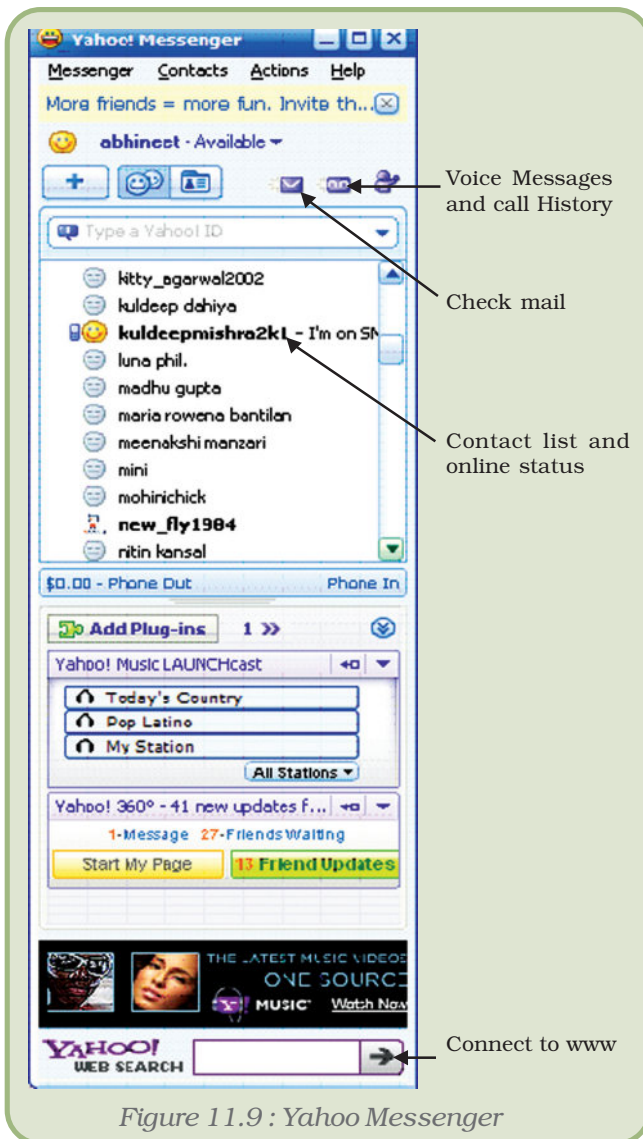- Keep security of the server computer very high.

Figure 11.9 : Yahoo Messenger

Voice Messages and call History

Check mail

Contact list and online status

Connect to www

Outcome

All the computers of the school are connected through network and Internet.

Project Title : Collaboration through Messenger.

There are many Messengers the Google, Yahoo etc. which provide the services requested like mail, chat, make voice call and send instant messages after logging in with your user account and password. Such utility is very useful in event management.

Illustration 1 : Yahoo Messenger

Invoke the Yahoo Messenger and enter login ID and password. After logging in the Yahoo Messenger, it will show the names of the persons highlighted who are on-line in your list as shown in figure 11.9 :
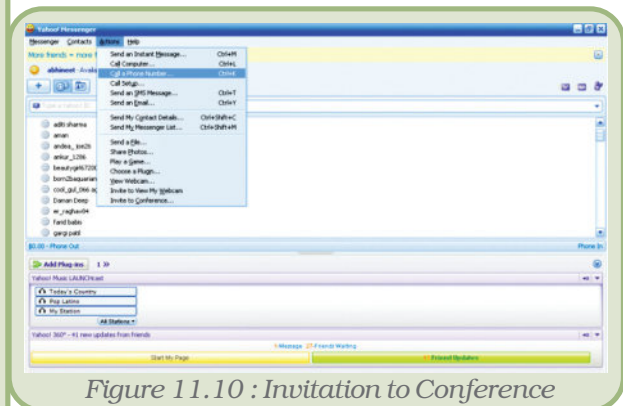

Figure 11.10 : Invitation to Conference

The different services are listed out in the **Actions** pull down menu. Move the mouse pointer on the particular action and click it to use it such as invite to conference as shown in figure 11.10.

After clicking you can start conferencing with your friends by typing the messages (Figure 11.11).

Choose another service such as **Send a file** from the action pull down menu (Figure 11.12).

Select the file from your computer by browsing the location of the file shown in figure 11.13 :
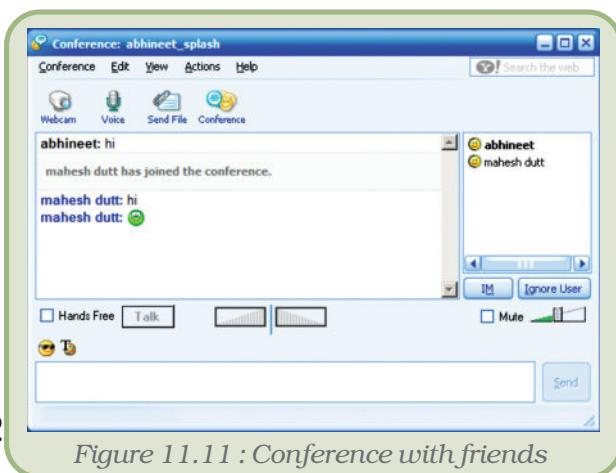

342

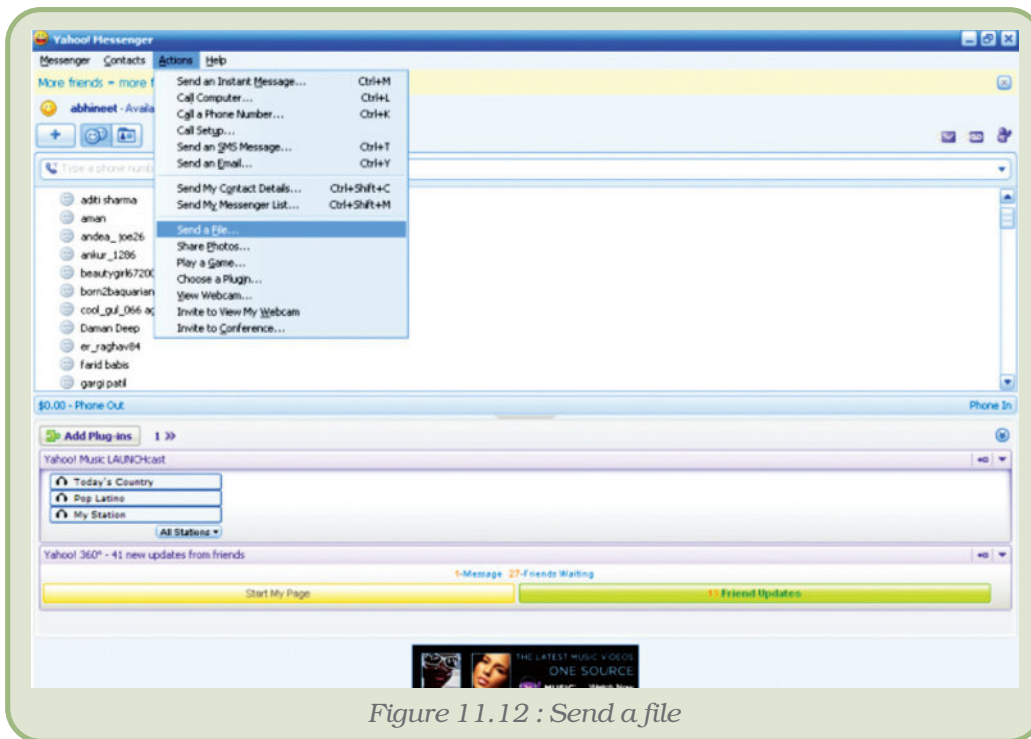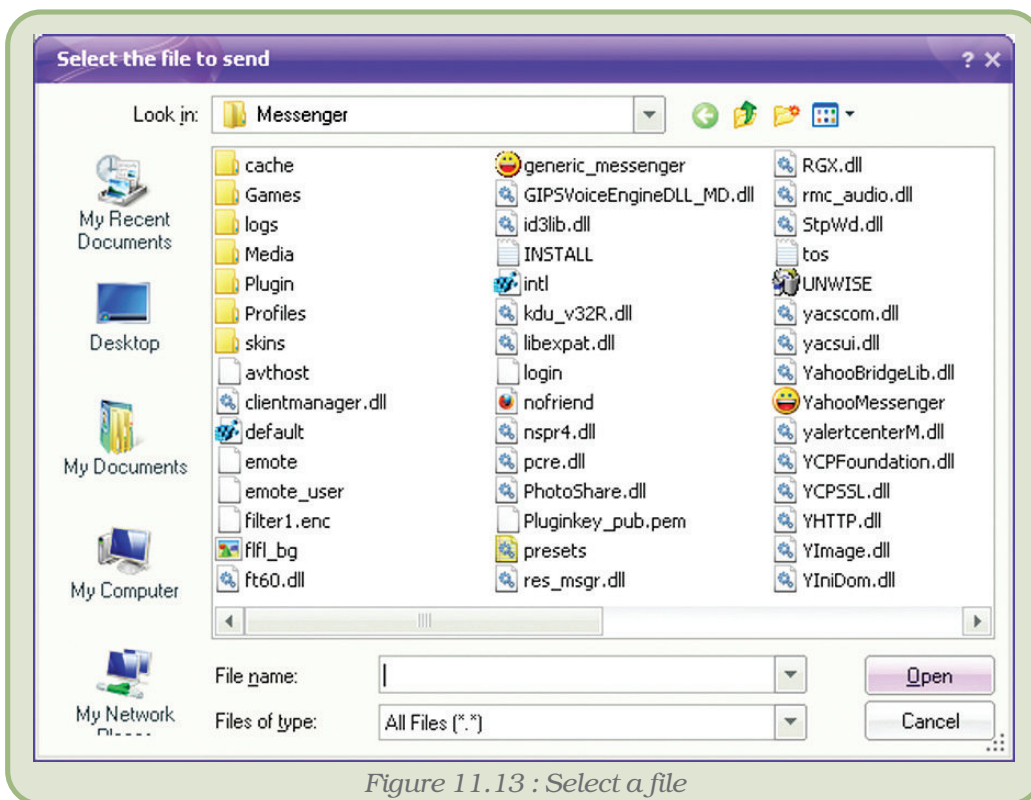*Figure 11.11 : Conference with friends*

Figure 11.12 : Send a file

Figure 11.13 : Select a file

To register your mobile device, click on **Preferences** of the Messenger pull down menu (Figure 11.14).
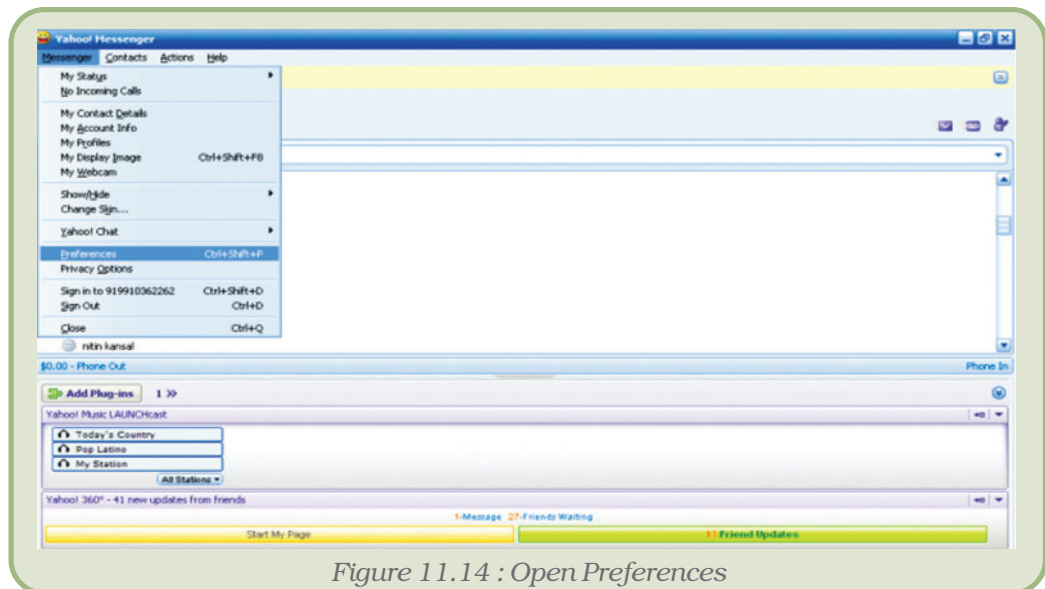
343

Figure 11.14 : Open Preferences

Go to General preferences and edit mobile device (Figure 11.15).
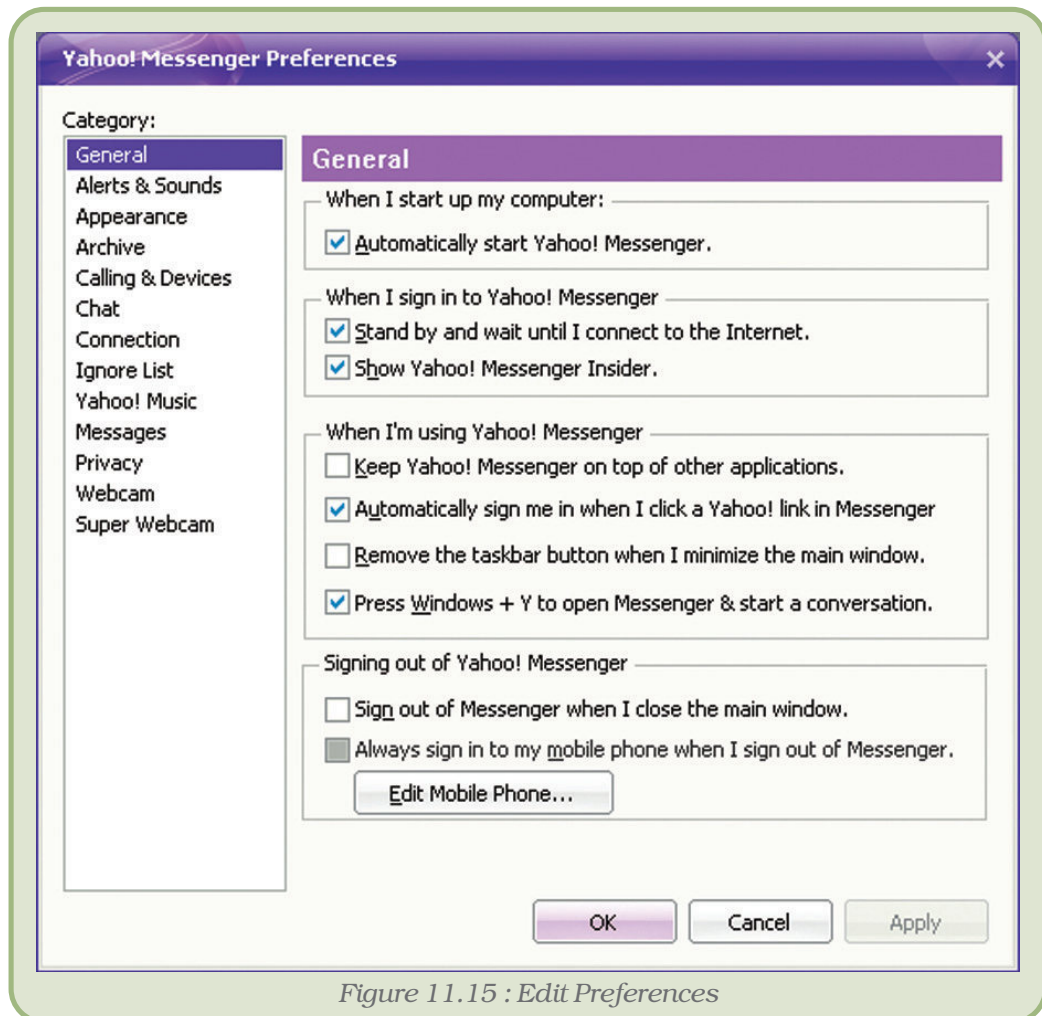

Figure 11.15 : Edit Preferences

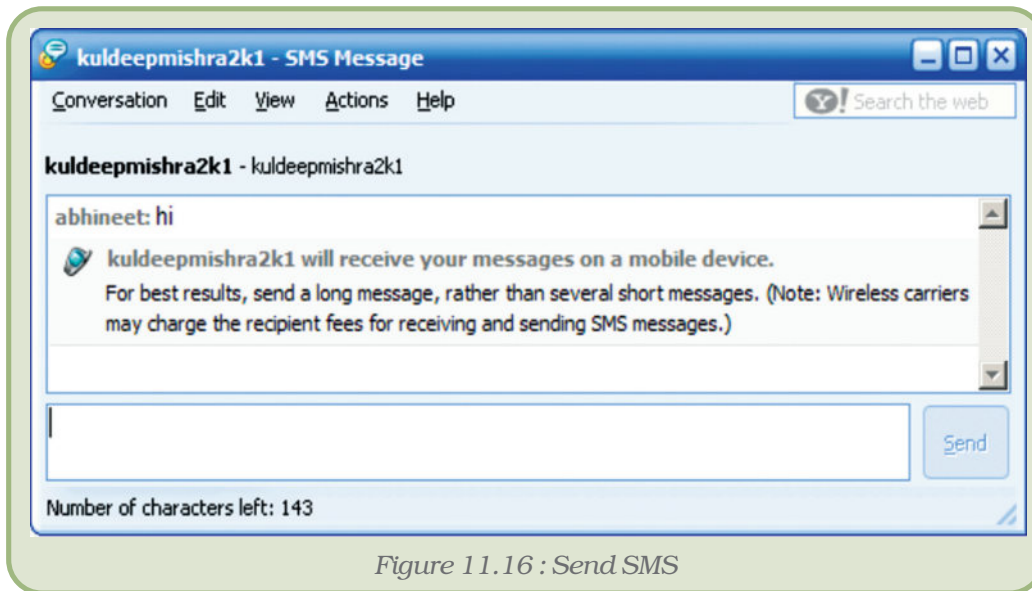Send SMS to your friend on mobile using the services in Yahoo Messenger (Figure 11.16).

Figure 11.16 : Send SMS

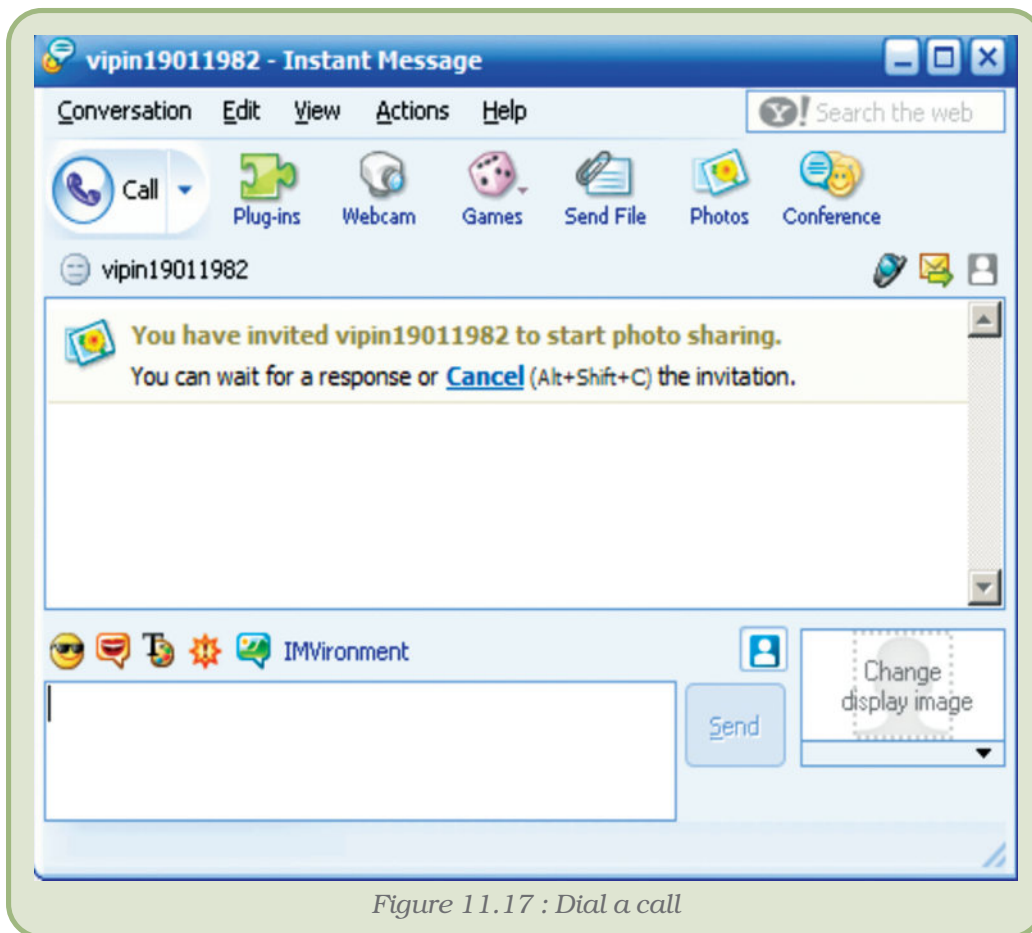Create groups of users for photo sharing (Figure 11.17).
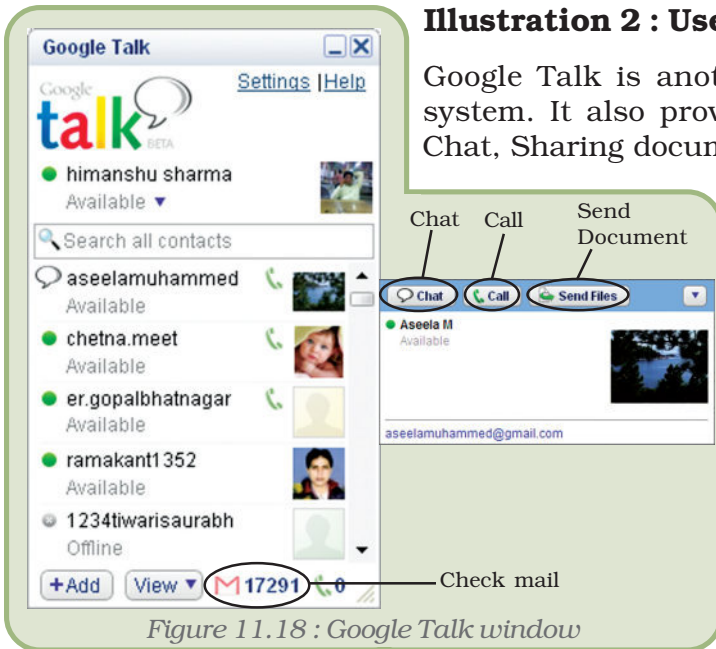

Figure 11.17 : Dial a call

345

Chat   Call   Send Document

Check mail

Figure 11.18 : Google Talk window

Illustration 2 : Use of Google Talk

Google Talk is another very popular Unified Messaging system. It also provides all the services such as Email, Chat, Sharing document, Voice Call and instant messages.

Invoke the Google Talk utility from your computer and enter the login ID and password. After logging in, the Google Talk window will show the list of friends logged as in figure 11.18 :

Instant Messaging and Chatting using Google Talk can be done as shown below in figures 11.19 and 11.20.

Making voice call using Google Talk can be done by clicking on call button.
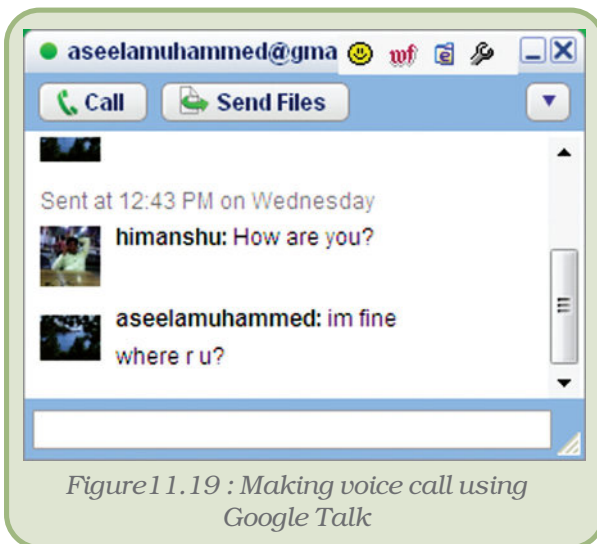
Figure 11.19 : Making voice call using Google Talk
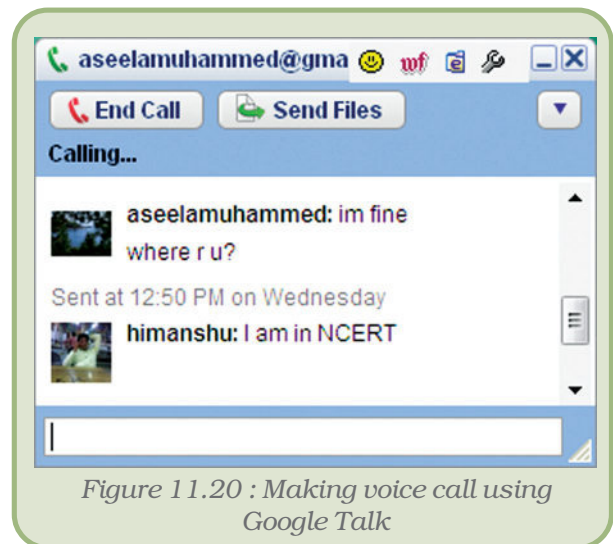
Figure 11.20 : Making voice call using Google Talk

Summary

- Project–based learning activities provide opportunities for students to work on problems in the real world.

- Modular approach is one of the *approaches to execute a project.*

- Projects can build students' abilities to set personal goals and standards of excellence.

- The success of the project depends on the whole team and not on an individual.

- Each team member should possess characteristics such as a Coordinator, Shaper, Worker, Specialist, Accelerator, Finisher, and Critics to become the role model.

- Teamwork is based on listening to others, sharing ideas and opinions with team-mates, giving equal respect to every team member, helping nature, ensuring the participation of each member in the discussion or working.

- Tracking changes is very useful in project work.

- Various techniques can be used to solve day to day problems in stages to achieve the proposed goals.

EXERCISES

SHORT ANSWER TYPE QUESTIONS

1. What is project based learning? What are its characteristics?

2. Comment on 'Projects can be executed only with teamwork' with the help of examples.

3. What is meant by modular approach to project based learning? Explain its various subtasks with the help of examples.

4. Explain different components of teamwork with the help of an example.

5. What are the benefits of working through projects?

6. What are the advantages of modular approach?

7. Describe the roles of different team members in a project.

LONG ANSWER TYPE QUESTIONS

1. Explain the steps involved in project based learning with a suitable example.

2. Explain with an example the benefits of CCT tools (like e-mail, Internet, track changes etc.) in implementation of a project.

3. Explain the use of Yahoo Messenger in event management with a suitable example.

MULTIPLE CHOICE QUESTIONS

1. A project is a — work that is carefully planned to achieve a particular aim.
    (i) Hard
    (ii) Collaborative
    (iii) Regular
    (iv) Simple

2. The most commonly used approach to execute a project is — approach.
    (i) Top down
    (ii) Modular
    (iii) Bottom up
    (iv) Sequential

3. An effort made by many people together in order to accomplish a task is called —
    (i) Teamwork
    (ii) Good work
    (iii) Hard work
    (iv) Best work

347

4. Teamwork meetings are conducted just for —
   (i) Listening to others
   (ii) Shouting at others
   (iii) Social gathering
   (iv) Amusement

5. Which of the following is not a component of a teamwork?
   (i) Listen to others
   (ii) Share with others
   (iii) Respect others
   (iv) Blame others

Activities

Write down the steps involved in execution of the following projects.

1. Event management

2. Literacy monitoring of a region

3. Preparation of a website for time table of a school

4. Establishing a wireless network of computers

CCT PROJECTS IN LOCAL CONTEXT

12

Objectives

After completing this Chapter, the student will be able to:

- *understand the meaning of e-Governance,*

- *describe National e-Governance Plan (NeGP),*

- *recognise the relevance of e-Governance from the governance point of view,*

- *comprehend the architecture of e-Governance,*

- *appreciate the evolution of e-Governance practices in India and*

- *identify some of the Government's e-Governance initiatives in local context.*

"Three years ago India was emerging as an IT superpower. Today, the country is handling the most sophisticated projects in the world"

- Bill Gates, 30 July 2004

Introduction

There are several services provided by the government to solve our day-to-day problems. With the growth of CCT the government has taken steps to use various computer communication tools and technology to modernise its working. It helps in providing the services more conveniently and a means for the public to easily avail these services. This affects the service provider (Government) and the recipient (the citizen) both. Most of these projects are driven by local needs to exploit CCT for better services, efficiency and transparency. Various state governments have implemented projects with reference to the local context.

This chapter will provide exposure to various initiatives taken by the government in implementing projects covering various aspects of our society.

12.1 NEED FOR e-GOVERNANCE

An organisation, be it government or private, will be left out of the race if it does not keep pace with the major global shifts towards increased

deployment of IT. In a fast moving world, the expectations of clientele will be naturally least tolerant to undue delays. Thus to sustain and survive, it is imperative for any organisation to reach out to its clientele and adapt to the changing paradigms by using technology. In any country, its government is a major service provider and answerable to its citizens in reaching out the benefits of its vast resources to them. The government's various initiatives in becoming techno-savvy constitute various components of e-Governance.

12.2 DEFINITION OF e-GOVERNANCE

e-Governance refers to the application of CCT to enhance the efficiency, effectiveness, transparency and accountability of informational and transactional exchanges at all levels. These exchanges can be:

- within the government, i.e. between government and government agencies of National, State, Municipal and Local levels,
- between citizen(s) and government.

The aim of e-Governance is to empower citizens through access and use of information.

12.3 EXPECTATIONS FROM e-GOVERNANCE

The services provided via e-Governance should be sustainable to cover all the segments of a society, at an affordable cost and customised to the local requirements. The services delivered by them should be reliable, authenticated, and timely, in a format and mode that is convenient to the users. It is estimated to provide complete services to meet the purpose.

m-Governance is the future of e-Governance services delivery, which will ensure inevitability and accessibility of e-governance services anytime, anywhere using mobile devices laptops, palmtops, mobile phones etc.

12.4 GOVERNMENT'S INITIATIVES

12.4.1 e-GOVERNANCE PROJECTS

Mostly, e-Governance projects are designed with various objectives such as providing easy access, extending access to un-served groups, introducing transparency, simplifying transaction procedures, minimising cost to citizens and to the government, increasing the government revenue, reducing the transaction time, offering new services, modernisation/adoption of best practices.

Implementation of these projects makes lives of common man better through technology implementation at national level in central government, at state level governments, at district level and finally at the village level.

National e-Governance Plan (NeGP)

The NeGP plan (2003-2007) was formulated by the Department of Information Technology (DIT), Ministry of Communications and Information Technology, Government of India to initiate and provide for long term growth of e-Governance in our country with the following vision :

"Make all Government services accessible to the common man in his locality, through common service delivery outlets and ensure efficiency, transparency and reliability of such services at affordable costs to realise the basic needs of the common man."

NeGP aims to change and improve drastically the way in which the Government provides services to its citizens and empowers them to demand convenient, cost effective and transparent services.

Figure 12.1 : e-Governance architecture under NeGP
Source: DIT, Ministry of Communication and Information Technology
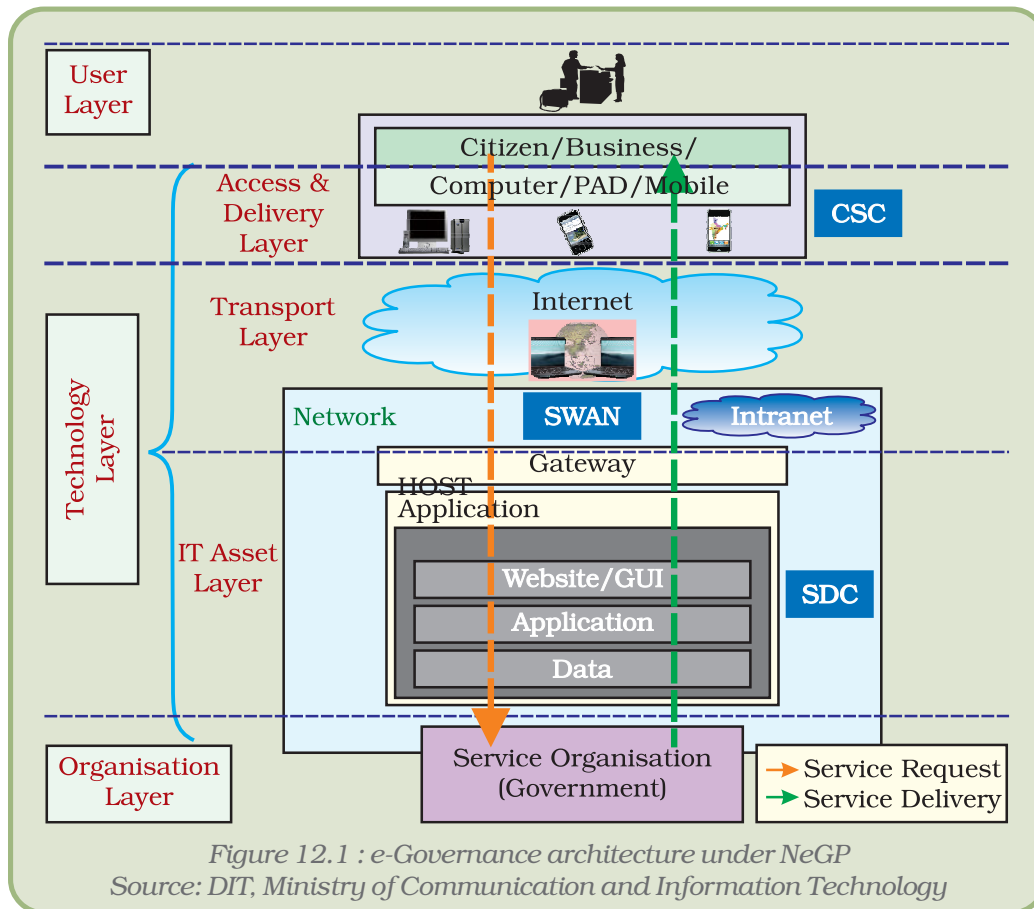
The architecture of NeGP consists of three layers: Organisation layer, Technology layer and User layer. The sharable infrastructure under NeGP is :

(a) Common Services Centres (CSCs) to facilitate services delivery electronically at the grass root level
(b) State Wide Area Networks (SWANs)
(c) State Data Centres (SDCs)

Technology Development for Indian Languages (TDIL)

Ministry of Information technology has taken initiative to start the TDIL programme. The aim of this programme is to develop and promote the use of Information processing tools and techniques for Indian Languages. Some of the objectives of the programme are to :

- facilitate human-machine interaction in Indian languages and multilingual knowledge resources.
- consolidate technologies developed for Indian Languages and integrate them to develop innovative user products and services like Multi-lingual Dictionaries, Encyclopedia, Gyan-nidhi Creative Writing System, Translation Support Systems, Text-to-speech and Speech Recognition System, Pocket Translator, Reading machine for the blind and the deaf portals.
- provide requisite technology development and bridging the existing technology gap in the Hindi speaking states of India.

12.4.2 e-GOVERNANCE PROJECTS IN LOCAL CONTEXT

There are several projects existing in different states in regional languages. The goal behind these projects is to make various public services and schemes available through technologies to a common man in their own understandable language. Through these IT based projects government tries to ensure the following:

- Enhancing awareness of IT enabled services
- Improving IT penetration
- Localised solutions
- Availability of IT learning material in local language
- Standardisation

12.4.3 MISSION MODE PROJECTS (MMPs)

NeGP implements a significant number of Mission Mode Projects (MMPs) and various projects at the central and state government level.

Central MMPs	State MMPs	Integrated MMPs
• Income Tax	• Agriculture	• e-BIZ
• Customs and Central Excise	• Land Records	• EDI(Electronic Data Interchange)
• Passports/Visa and Immigration	• Transport	• India Portal
• MCA 21	• Treasuries	• Common Service Centres (CSC)
• National ID	• Commercial Taxes	• EG Gateway
• Pensions	• Gram Panchayats	• e-Procurement

• e-Office	• Municipalities	• e-Courts
• Banking	• Registration	
• Insurance	• Police	
	• Employment Exchange	
	• e-District	

Source : DIT, Ministry of Communication and Information Technology

12.5 APPLICATION AREAS OF e-GOVERNANCE PROJECTS

Some of the departments and services where e-governance is applied are:

- **Public Grievances :** Electricity, Water, Telephone, Ration Card, Sanitation, Public Transport, Police.

- **Rural Services :** Land Records, Below Poverty Line (BPL)/ Economically Weaker Section (EWS) Families.

- **Police :** FIR Registration, Lost and Found valuables and persons.

- **Social Services :** Pension, Old Age, Widows, Ex-gratia Scheme, Acquisition/Rehabilitation and Compensation, Registration of Licences and Certificates, Ration Card, Birth Certificate, Death Certificate, Domicile Certificate, Caste/Tribe Certificate, Arms Renewal, Registration of Documents, School Registration, University Registration, Motor Vehicle Registration, Driving License.

- **Public Information :** Employment Exchange Registration, Employment Opportunities, Examination Results, Hospital Availability/Services, Railway Time Tables, Airline Time Tables, Road Transport Time Tables, Charitable Trusts, Government Notifications, Government Forms, Government Schemes.

- **NEWS Services :** Civil supplies, Old Age Pension, Widow Pension, Handicapped Pension/Services, Ex Gratia Payment.

- **Agriculture Sector :** Speeds Information, Pesticides, Fertilisers, Crop disease, Weather Forecast – short range/District wise, Market Price.

- **Utility Payments/Billing :** Electricity, Water, Telephone.

- **Commercial :** Taxation and Return Filing, Income Tax, Corporate Tax, Custom Duty, Central/State Excise Duty, Sales Tax, House Tax, Property Tax, Octroi, Road Tax, Company Returns.

- **Government :** Electronic Procurement, Education University Model for e-Governance.

12.6 IMPLEMENTATION OF VARIOUS PROJECTS

Table 12.1 : Some of the projects implemented by various state governments

State/Union Territory	Initiatives covering departmental automation, user charge collection, delivery of policy/programme information and delivery of entitlements
Andhra Pradesh	e-Seva, CARD, VOICE, MPHS, FAST, e-Cops, AP online—One-stop-shop on the Internet, Saukaryam, Online Transaction processing, India Health Care Project, Bhu Bharati
Assam	e-Suvidha, Dharitri, ASHA, Samvad
Bihar	Sales Tax Administration Management Information, Flood Monitoring by Satellite Pictures, SCORE, VAT, Information Computerisation, ELECON
Chattisgarh	Chhattisgarh Infotech Promotion Society, Treasury office, e-linking project
Delhi	Automatic Vehicle Tracking System, Computerisation of website of RCS office, Electronic Clearance System, Management Information System for Education, Delhi Slum Computer Kiosks Project, Smart Card Based Driving Licence and Vehicle Registration Certificate etc.
Goa	Dharani Project, Municipal Administration Software
Gujarat	Mahiti Shakti, request for Government documents online, Form book online, G R book online, census online, tender notice, Jan Seva Kendras, Chiranjiv Yojana, Nirmal Gujarat
Haryana	Nai Disha, Janani Suvidha Yojana
Himachal Pradesh	Lok Mitra, Himachal Buses: Transport Tracker
Jammu and Kashmir	DACNET
Jharkhand	Treasury computerisation, Vahan, Sarathi, Land record Information System
Karnataka	Bhoomi, Khajane, Kaveri, Krishi Marata Vahini, Gram Swaraj Project
Kerala	e-Srinkhala, RDNet, Fast, Reliable, Instant, Efficient Network for the Disbursement of Services (FRIENDS), Akshaya, Asraya, SWIFT, PEARL
Madhya Pradesh	Gyandoot, Gram Sampark, Smart Card in Transport Department, Computerisation MP State Agricultural Marketing Board (Mandi Board) etc., Head Start, Rogi Kalyan Samiti's

Maharashtra	SETU, Online Complaint Management System–Mumbai, Telemedicine Service in Pune Primary Health Centres, Warana-Wired Village Project
Orissa	GRAMSAT–ORISSA Computer Application Centre
Punjab	Punjab Govt. On-line, e-procurement, e-district
Rajasthan	Jan Mitra, RajSWIFT, Lokmitra, RajNIDHI, e-Mitra, SARATHI
Sikkim	Sikkim Treasury Online, Payroll Information System
Tamil Nadu	Rasi Maiyams–Kanchipuram, Application forms related to public utility, tender notices and display, Reginet, E–Registration Star
Tripura	Gramoday, Hospital Management System, Vahan, e–Suvidha Kendras
Uttarakhand	Saksham, Aarohi, Computerisation of Urban development department
Uttar Pradesh	Lok Vani, e–Suvidha, Bhulekh, Koshvani, PRERNA
West Bengal	Tele Medicine-Midnapur, Smart Card based Driving License and Registering Certificate using Sarathi, Computerisation of Government Departments, Kolkata Police Intranet and Computer Network
North-Eastern States : Arunachal Pradesh, Manipur, Meghalaya, Mizoram and Nagaland	Community Information Centre. Forms available on the Meghalaya website under schemes related to social welfare, food civil supplies and consumer affairs, housing transport etc., e-Suvidha, Sarathi, Vahan, Transport Department Computerisation
Andaman and Nicobar Islands	Dweep Bhumi
Chandigarh	e-Sampark
Dadra and Nagar Haveli	Suvidha
Daman and Diu	Suvidha
Lakshadweep	PORTNET, Ever Alert, Web Enabled Entry Permit Management System, Integrated e-Governance solution to Lakshadweep Electricity Department
Puducherry	Gazettes on web, ICT for fishermen community

Source from DIT, Government of India

> ### *Summary*
>
> - With the advent of CCT, the Government of India has taken steps to use various technologies to modernise its functioning.
>
> - Various State governments also have implemented e-Governance Projects in reference to local Contexts.
>
> - e-Governance is the application of information and communication technologies to transform the efficiency, effectiveness, transparency and accountability of informational and transactional exchanges.
>
> - NeGP implements a number of Mission Mode Projects (MMP's) both at the central and state levels to create a citizen-oriented environment for e-Governance.
>
> - TDIL aims at promoting the use of IT tools for providing services in Regional Indian languages.
>
> - The goal behind projects in local context is to make various public services and schemes available to common man in their own understandable language.

EXERCISES

SHORT ANSWER TYPE QUESTIONS

1. Define e–Governance.

2. What is the need for e–Governance?

3. What is m–Governance?

4. Expand the following abbreviations:
   - CSC
   - NeGP
   - SWAN
   - SDC
   - MMPs

5. Name some on-going e–Governance projects?

LONG ANSWER TYPE QUESTIONS

1. What do you mean by e–Governance and what are its components?

2. Describe a few recent e–Governance projects implemented by your State/UT government?

3. What are the recent initiatives taken by government for implementing e–governance?

4. Elaborate the application areas/departments in which e-Governance can be applied.

5. What are the expectations of recipients from the services delivered by m–Governance?

6. Explain NeGP architecture with the help of a diagram.

7. Describe some initiatives taken by Government to promote CCT tools in Indian languages.

8. Discuss the applications of e–Governance Projects in local context with examples.

EMERGING TRENDS IN CCT

13

Objectives

After completing this Chapter, the student will be able to:

- *appreciate and describe the emerging trends in the field of computer software, computer storage and processing and types of storage devices,*

- *understand and interpret the kind of technology that will drive the next generation computer controlled devices and*

- *explain Nanotechnology.*

"Logic will get you from A to B. Imagination will take you everywhere."

Albert Einstein

Introduction

By now you have come out of the conventional ideas of a computer whether in regard to its physical appearance or its functionalities and seen the multifarious fields of activities in which the principle is operative irrespective of the physical appearances of the device. With electronic devices becoming ubiquitous, it is important to know and understand the latest trends and anticipate what is coming in the near future. It is a process of continuous evolution, constantly updating and improving in efficiency and functionality.

Not only the computer system has become faster, compact and inexpensive but associated communication system has also become affordable, reliable and user friendly.

The processing speed of the computers is getting nearly doubled and every year a new processor is released. Still research is going on to look for different avenues for improving computer processing using improved materials, nanotechnology, multi core processors, quantum computers, etc.

The emerging trends in the field of CCT is also focussed on research and development on energy efficient systems, wireless and optical communication, biometric, security system, multimedia and on demand software. This chapter will broadly focus on the following fields.

1) Hardware
2) Semiconductor
3) Communication system
4) Quantum Computing
5) Software
6) Nanotechnology

13.1 EMERGING TRENDS IN HARDWARE

The advancements in CCT are bringing us devices in miniaturised forms like digital cameras, cell phones, iPods and flash disks etc.

13.1.1 STORAGE DEVICES

In computer storage, the trend is towards increased capacity in terms of memory and processing speed on one side and reduction in size and cost on the other side.

Today, the type of data is no more merely text documents, it is in multimedia form which require more memory space for storage. The research in this field led to development of storage devices like iPod, Pen-drives, flash disks, DVDs which have capacities to store data upto a maximum of 50GB as on today. An iPod can store music, videos, documents, pictures and games. Their storage capacity varies from 1GB to 80GB.
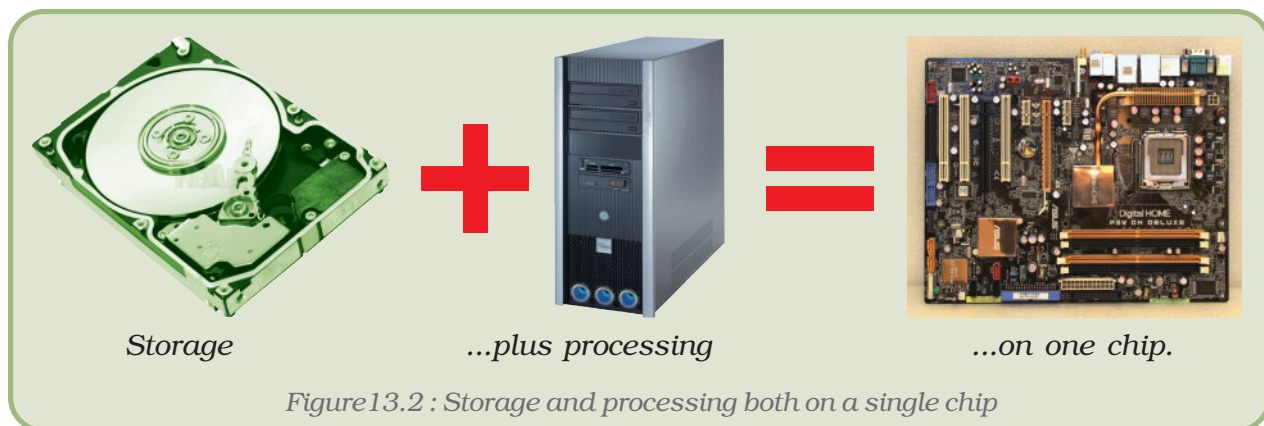

Figure 13.1 : An iPod

Hard Disk

We have seen tremendous growth in the hard disk capacities over years which is expected to grow even further. Hard disk drives have become faster, with faster seek time, larger cache sizes, and higher interface speeds. One of the most important parameters for improving performance is the higher spindle speed, effectively improving the internal data rate and reducing latency. Nanotechnology has played an active part in reducing the size of hard disks and in the increase of their capacities.

Hybrid Hard Disk Drive (HHDD) is a newer technology where the conventional disk drive is combined with non-volatile flash memory, of typically 128MB or more to cache data during normal use. The data is initially stored in non-volatile memory before permanently storing it in the hard disk. Enterprise HDDs are specifically designed for mission-critical applications such as core servers and large-scale storage systems. The first hybrid hard disk drives were 2.5 inch drives for notebooks.

Whole Disk Encryption/Full Disk Encryption (FDE) is a new technology (hardware or software) where data is encrypted before storage. This prevents unauthorised access or retrieval of data.

Micro Electro Mechanical Systems [MEMS]-based storage is a new technology being developed as a new age storage media due to its attractive features such as small size, shock resistance, and low-power consumption. MEMS-based storage is anticipated to be widely used for mobile consumer electronics.

Storage   *...plus processing*   *...on one chip.*

Figure13.2 : Storage and processing both on a single chip

Redundant Array of Inexpensive Disks (RAID)

Redundant Array of Inexpensive Disks (RAID) is a way of storing the same data in different places redundantly on multiple hard disks. This helps to restore the copy of the data even if one of the disks fails. The technology allows replacing the faulty disk without shutting down the system.

Emerging trends in RAID allows us to save expensive rack space and reduce co-location costs.

3-D Optical Data Storage

3D optical data storage is a new technology where data is stored in multiple layers in the optical disc. Laser beam is used for reading and writing the data from/to an optical disc. The disc contains many layers of data, each at a different depth in the media and each consisting of a DVD-like spiral track (Figure 13.3).
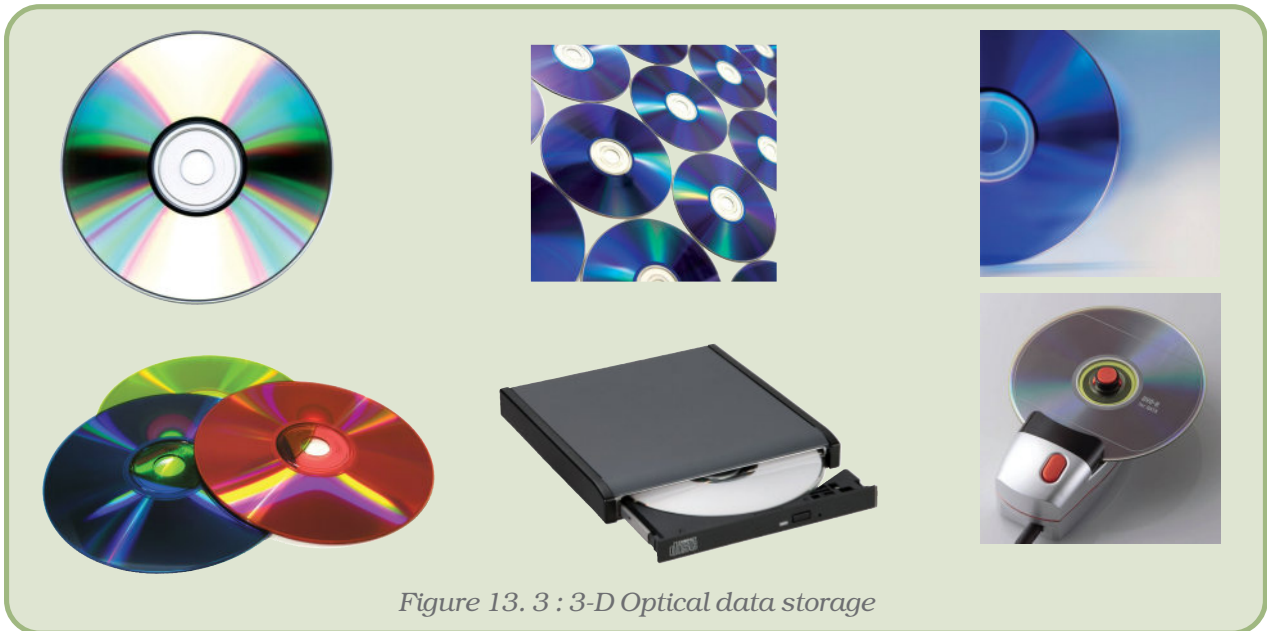

Figure 13. 3 : 3-D Optical data storage

Holographic Memory

Holographic Memory is quite similar to 3-D Optical Data Storage. Here data is recorded through full depth of the media instead of only on the surface. Thus it can record and read millions of bytes of data with a flash of light. High density Optical disc is an example of holographic storage.

Network Storage

Network storage allows consolidating all data spread over various parts of the organisation into one location, from where it is accessed by multiple applications across the network simultaneously. There are two technologies which are being used - SAN and NAS.

Storage Area Network (SAN)

Storage Area Network (SAN) is an architecture where the different secondary storage devices like hard disk arrays, tape drive etc. are attached to remote computer storage devices and all the servers are connected to the SAN through the SAN switch and the servers can access data as local disk drives. The remote storage devices are shared by multiple servers simultaneously (Figure 13.4).

Figure 13.4 : Storage Area Network (SAN)
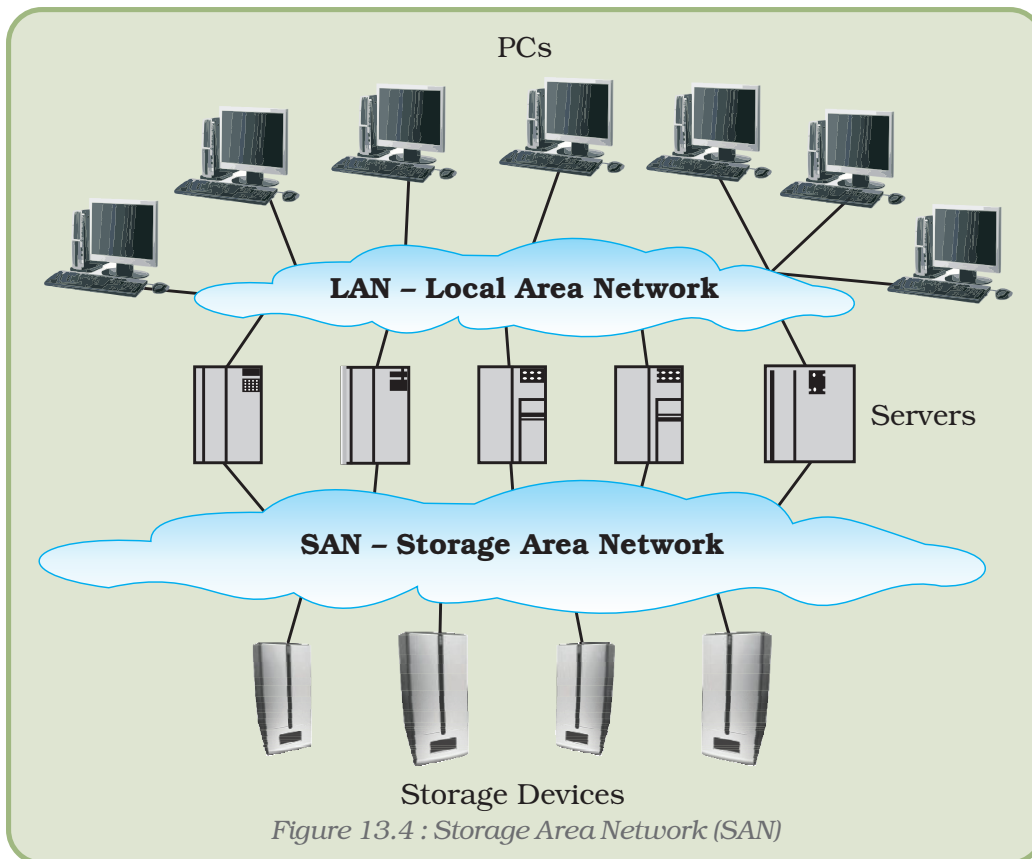
Network Attached Storage (NAS)

Network Attached Storage (NAS) uses remote computer with storage devices which are connected through TCP/IP network as shown in Figure 13.5. It uses file-based protocols such as Network File System or Common Internet File System (CIFS). All the servers and storage devices are connected through LAN or WAN.

Figure 13.5 : Network Attached Storage (NAS)

363
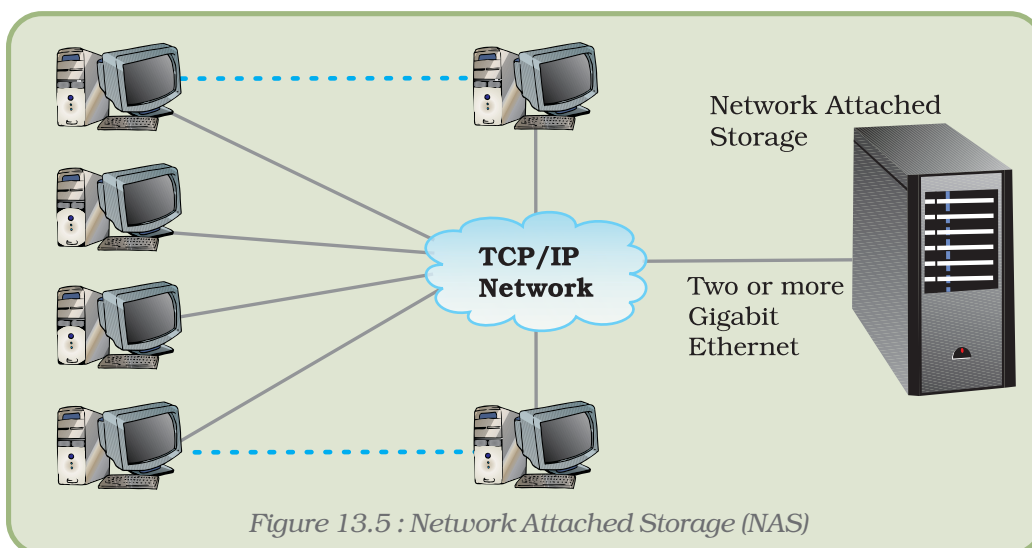
Online storage options

There are several companies which provide online storage to enable the user to store data on their servers. These services are offered to the Internet users freely but with limited storage. Network and distributed file systems, and Internet-based data centres are some examples of online storage implementations.

Peripheral/Interface

Universal Serial Bus (USB) is used for connecting peripherals to PC. The popularity of USB is due to its low cost, ease-of-use, and small connector size. It is easy to add USB support to any peripheral. Latest USB trends are discussed in the following section.

USB in embedded systems

USB has a vast number of applications in embedded systems like cell phones, PDAs, digital cameras, printers, and set top boxes.
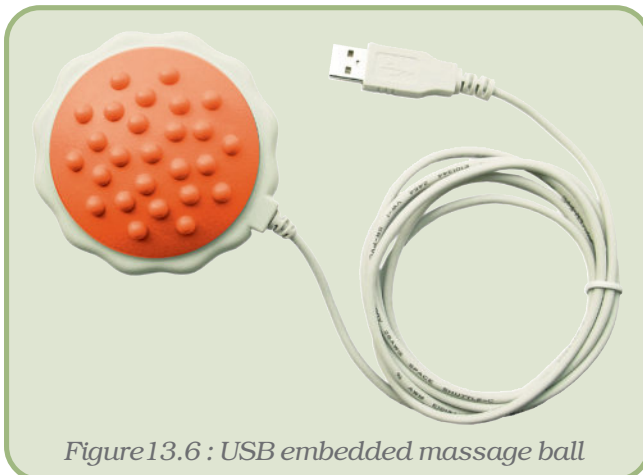

Figure 13.6 : USB embedded massage ball

Wireless USB

Wireless USB allows low-speed devices to connect without wire, but appear to the host as if they are connected over wired USB. Wireless USB (Figure 13.7) is an ideal solution for simple cable replacement applications.

13.1.2 MICROPROCESSOR

A microprocessor is a multipurpose programmable logic device that reads binary instructions from a storage device called memory, accepts binary data as input and processes data according to those instructions and provides results as output.

The conventional basic function of a single microprocessor has been changed to more complex microprocessor architecture. Instead of single microprocessor, packages of multiple microchips and processors are released which fulfil and meet the performance requirement for a computing system for a particular application. These packages are simply installed into standard interface on motherboards.

Figure 13.7 : Wireless USB

Similarly the **Core 2** processor refers to two separate dual-core dies (CPUs) – next to each other – in one quad-core package.

Xeon processors refer to many families of Intel's multiprocessing CPUs – for dual-processor (DP) and multi-processor (MP) configuration on a single motherboard. They are used for server and workstation computers. It has been maintained over several generations of x86 and x86-64 processors. The *Xeon* CPUs generally have more cache than their desktop counterparts, in addition to multiprocessing capabilities.

Figure 13.8 : Intel dual core 2

13.2 EMERGING TRENDS IN SEMICONDUCTORS

Use of innovative technologies like Silicon-On-Insulator (SOI), Complementary Metal-Oxide-Semiconductor (CMOS), capacitor-less memory, Micro-Optic-Electro-Mechanical-System (MOEMS), III-V compound materials-on-insulator and others have improved the performance and also reduced the size of consumer electronic devices.

Apart from the development of innovative designs, researchers are also exploring materials other than Silicon to develop high speed electronic chips. One such example is Graphene, which is a form of pure carbon where – a single layer of carbon atoms are arranged in a honeycomb lattice. It could allow electrons to process information and produce radio transmissions 10 times better than silicon-based devices. Usage of Graphene will produce faster and more powerful cell phones, computers and other electronic devices, since the mobility of electrons is more than any existing semiconductor materials, like Silicon, GaAs and carbon nanotubes.
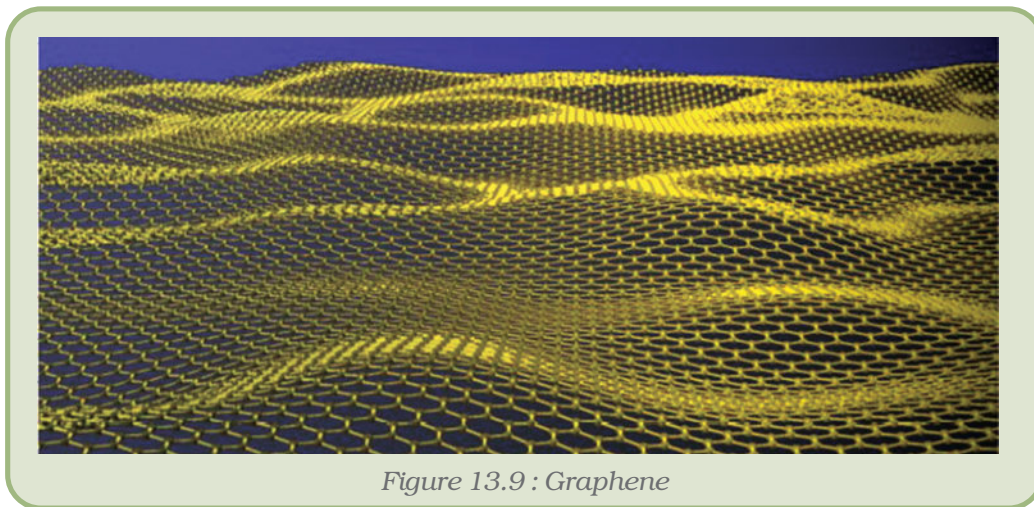

Figure 13.9 : Graphene

Another key area where extensive research is being carried out is the development of application processor of 3D graphics and RAM memories. Research is going on to develop multiple semiconductors into single packages.

13.3 EMERGING TRENDS IN COMMUNICATION SYSTEMS

In communication the trend is to develop integrated multimedia systems such as VoIP (Voice over Internet Protocol), IP TV, Video Conferencing, Video on Demand etc. The emphasis is to upgrade the legacy communication system that can support multimedia applications.

Apart from these, technologies are being developed to provide cost effective networking solutions and to extend the network to end user premises. Examples in this area include multi service broadband network, DTH (Direct-To-Home), Fiber-to-home etc.

13.4 EMERGING TRENDS IN QUANTUM COMPUTING

Since the invention of Optical fiber and Lasers, use of light in computers and communication systems is being increasingly explored. Quantum computing means using light instead of electricity for data storage and transmission. Optical communication system had brought revolution in the transmission industry and communication network.

Efforts are being made to develop Silicon Lasers to replace silicon electronic chips. This would not only improve the speed, but also give energy efficient devices. Research and development is going on to design optoelectronic devices using optical component with silicon.

The research in the field of photo-electronics would help in developing cost effective next generation devices that would use both electrons and photons to process information. The development of photo-electronic devices is being done with collaborative effort of engineers and scientists from various fields as optics, chemistry, electronics and material science.

Apart from optical communication systems, optical devices such as optical mouse, optical hard disk, optical sensors etc. have been/are being developed to meet consumer electronic sectors.

13.5 EMERGING TRENDS IN SOFTWARE

Computer software and hardware go hand in hand and improvements in one lead to improvement in another. The design of the computer software has seen a paradigm shift – from structured programming to object oriented programming. Open source software is gaining popularity for its readiness to be extended by anyone and applied on any device. Moreover, they are freely available as they are developed by a community of developers.

Initially, software was produced largely by academics and corporate researchers working in collaboration and was not itself seen as a commodity. Variety of software ranging from that used in a mobile phone to the software for designing an aeroplane are available these days.

13.5.1 OPERATING SYSTEM

The Operating Systems which do not provide sufficient flexibility to the application developers like UNIX are beaconing obsolete. The latest trend is to design and develop customised application specific Operating Systems which can detect hardware failure and have auto-recovery features.

13.5.2 APPLICATION SOFTWARE

Software development has become a more systematic process where a great emphasis is laid on finding out the user requirement, preparing a process model, demonstrating an early version to its customers (prototyping) and on trying to ensure that the software is free of errors. Moreover, an incremental software development process is followed where a project proceeds in small and manageable steps.

An important trend in software is the use of component driven software development. Here, instead of writing programs from scratch over and over again, software developers can select components from comprehensive libraries of reliable and well documented software components and combine them together to carry out the required purpose.

The latest market trend is to develop Commercial-Off-The-Self (COTS) packages which can meet upto 50% system functionalities and can be used to develop customised applications (which meet the rest of the user requirements). The applications are developed by using existing reusable components of COTS packages and combining them with the new requirement. Such components are required to be useful, easy to use and interoperable which can work across networks, the Internet and with web browsers. Many such packages such as GIS, Simulators, CAD/CAM, SAP etc. are available in the market and applications are being developed using these COTS packages.

There are enormous number of application software catering to our needs. The capabilities are increasing day by day as per needs. Grossly we can categorise them according to their use as :

* Educational and communication software
* Design, media and simulation software
* Office automation and process mangement software
* Control and analysis software

367

Educational and Communication Software

More and more educational software on topics simple to complex, ranging from learning alphabet to experiments on genetics are being developed for learning purposes. In this category, the **Edutainment Software** are getting more and more popular because the presentations are done in multimedia form and have educational value in entertaining way. Thus learning becomes a stress free and enjoyable exercise to the user. A few indigenous multimedia software are Milestone in Genetics, Locus, Ray Optics, etc.

The **Learning Management System** (**LMS**) are software which manage user learning interventions. Most of the LMSs are web-based and go far beyond conventional training records management and reporting (e.g. ATuTor, Moodle, Brihaspati-II, Enrich). They cater to the needs of the user in training workflow (e.g. user notification, manager approval, waitlist management), the provision of on-line learning, on-line assessment, collaborative learning (e.g. application sharing, discussion threads), training resource management like instructors, facilities, equipment. They are used by regulated industries and also educational institutions. They are based on a variety of platforms like Java/J2EE, .net and PHP and use a database as backend.

Design, Media and Simulation Software

The advancements in CCT have not only resulted in higher processing speed, increased storage capacity and reduced price but also has brought out variety of high resolution output devices capable of displaying many millions of colours and shades and serving to a particular purpose.

The **Media development software** caters to the needs of people who generate print and electronic media, usually in educational and commercial settings. CatGrab (a digital camera software) for Windows, Track 'n share your GPS adventures!, Ashampoo 3D CAD Architecture are some media development softwares used.

Using **Image organiser** one can resize, tag digital images, organise pictures into albums by drag and drop, can export the pictures for external use (by e-mail or print). Some examples are Falco Icon Studio 4.8, Tif Joiner etc.

Graphic art software are primarily used for graphic design, multimedia development, specialised image development, general image editing and accessing graphic files. Ashampoo Photo Optimiser FREE, Ashampoo Photo Commander, ACX Diashow XL are such kind of software.

Image editing software allows us to manipulate and edit scanned images e.g. FastStone image viewer, Falco Icon Studio 4.8, Adobe Photoshop, PhotoPlusSE. Here by using Adobe Photoshop, a coloured picture (Figure 13.11) has been developed from black and white picture (Figure 13.10).

Figure 13.10 :
Black and white
picture

Figure 13.11 : Use of image editing software

Video editing software provides facility to import and export video, cut and paste a portion of a video clip, add special effects and transitions; encode the video for creation of a DVD, Web video, mobile phone video and synchronise the audio with the video which accompany the video (e.g. Mobilevideo For iPod 3.6 b28, DVD to Video Converter etc.)

Digital Audio Workstation (DAW) allows us to manipulate recorded sounds, much like a word processor manipulates entered text.

Music sequencer (also **MIDI sequencer**) is package of software and hardware designed to create and manage electronic music. Sequencer these days is synonymous with software but some hardware synthesizers and almost all music workstations include a built-in MIDI sequencer. Way back the music sequencers did not have the ability to record audio, they only managed "remote-control" information (such as "note on" and "note off" events) and sent to electronic musical instruments to produce the audio output. Now a days sequencers also feature audio editing and processing capabilities. Building Blocks, Space Toad MIDI Sequencer 1.1.4 are Music sequencers.

Diagramming software are used to produce graphical diagrams such as a flowchart, circuit diagram, network diagrams, technical drawings, tools etc. They allow the users to express the information in the form of a diagram. Diagram Designer 1.21, EDraw Max, SmartDraw, Dia (GPL license), OmniGraffle, Microsoft Visio, Inspiration, Concept Draw 7 are examples of Diagramming software. The flowchart in Figure 13.12 has been made using Diagram Designer which is a Diagramming software.
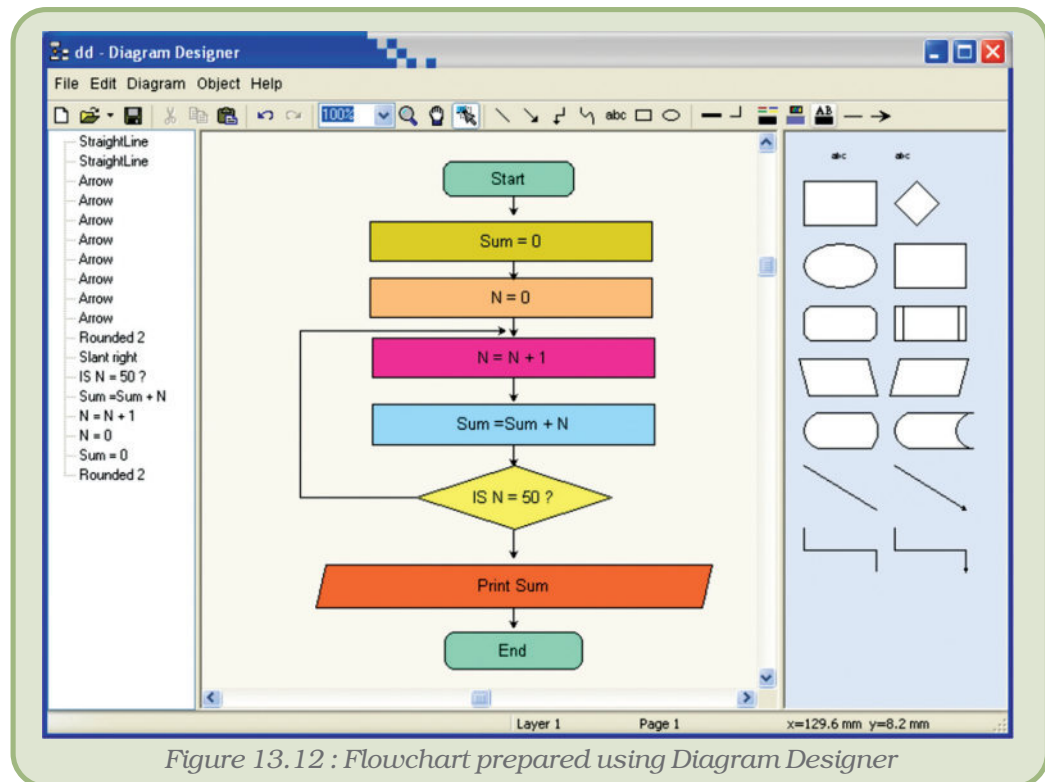
369

Figure 13.12 : Flowchart prepared using Diagram Designer

Computer Aided Design (CAD) software are used to create two and three-dimensional architectural design. Architects and designers usually make use of it in drafting and designing bridges, buildings, etc. and have been found to be very useful as instead of redrawing the whole plan one can modify or redraw with just a few clicks of mouse. Autodesk Streamline, AutoCAD, DWGcolumn are a few such software. Use of CAD

- saves time in making and editing the drawings
- enables the designers to lay out and develop work on the screen
- provides print out of the designs
- reduces the design cycle and over all product development costs
- stores the designs for future purposes in digital form thus occupying practically no space.

Simulation software simulates physical or abstract systems for learning, research, training or entertainment purposes. The **Social simulators** are used for learning the managing of socio economic aspects (e.g., crisis management, cooperation, competition, markets, social network dynamics, etc.). Battlefield simulators are used as war games, they work on theories of warfare. Training is done without the actual hostilities. Since the scope of war is not limited to military only but also political and social too, many governments make use of simulation, to test and refine their military and political policies. The **Flight simulators** are mainly used for training of pilots by providing them varying degrees of realism. These are video games or full-size cockpit replicas mounted on hydraulic (or electromechanical) systems,

controlled by the computer technology. The advantage is that training can be done in risk free situation. ORBITER is a free flight simulator that goes beyond the confines of Earth's atmosphere. The **Scientific simulators** present data to users in pictorial forms based on some model; which enhances the understanding of relationship among the variables/data being presented. Certain important results are derived.

Office Automation and Process Management Software

There are software designed to streamline and automate different types of work processes. In commercial and business environment a single software can process loan, mortgage, insurance, claim processing, etc. The software can automate office routine, administrative processes. Accounting software e.g. Tally**,** GnuCash, Turbocash, are used to record and process financial transactions within functional modules such as accounts payable, accounts receivable, payroll and trial balance in a company. In short one can keep track of the expenses, purchases and incomes etc.

Tax preparation software guides us to calculate taxes and fill out tax returns and thus saves us from the complex procedure of calculations.

A transportation software is used in logistics, trucking, supply chain, and automating dispatching, operations and financial systems within an organisation.

Control and Analysis Software

There are software which control the devices by presenting data in the desired format like images or chart etc. Some of the softwares coming under this category are:

- Medical software
- Mapping software

Medical software

It is a significant branch of software engineering. Many medical devices which are in use today to control or monitor patients are predominantly controlled by software. Given below are types of medical software :

- Monitors interpreters : use software to interpret the data recorded by sensor about the heart rate, blood pressure, breathing rate of patient and display it in a desired way on the monitor.
- Medication pumps interpreters: With the help of software, these devices are programmed to pump at a certain rate, desired amount of blood, saline solution, medication etc. to the patient.
- Analysis software : Many devices, such as CAT scanners, measure raw data that a common man cannot interpret. The analysis Software creates images by reinterpreting this data that the doctors can read and understand.
- Medical Informatics : It is concerned with the business and informational aspect of medicines.

371

Mapping software

They are also called as Geographic Information System (GIS). They make use of combination of capabilities of programs like database management, graphics and spreadsheet etc. for displaying data graphically, in tabular form, map etc. One can see various data and its patterns and relationships among the data w.r.t. geographical locations. Based on these presentations certain important political, social, educational decisions and predictions (e.g. weather, markets, etc.) can be done which can be useful to farmers, fishermen, professionals, corporates, politicians etc.. Earthquake 3D, ArcGis, GeoNetwork OpenSource, Key Indicator Data System (KIDS) etc. are some examples. The map (Figure 13.13) showing Teacher pupil ratio at primary stage 2002 has been prepared using Geo Media Professional 4.

Figure 13.13 : Presentation prepared using Geo Media Professional 4
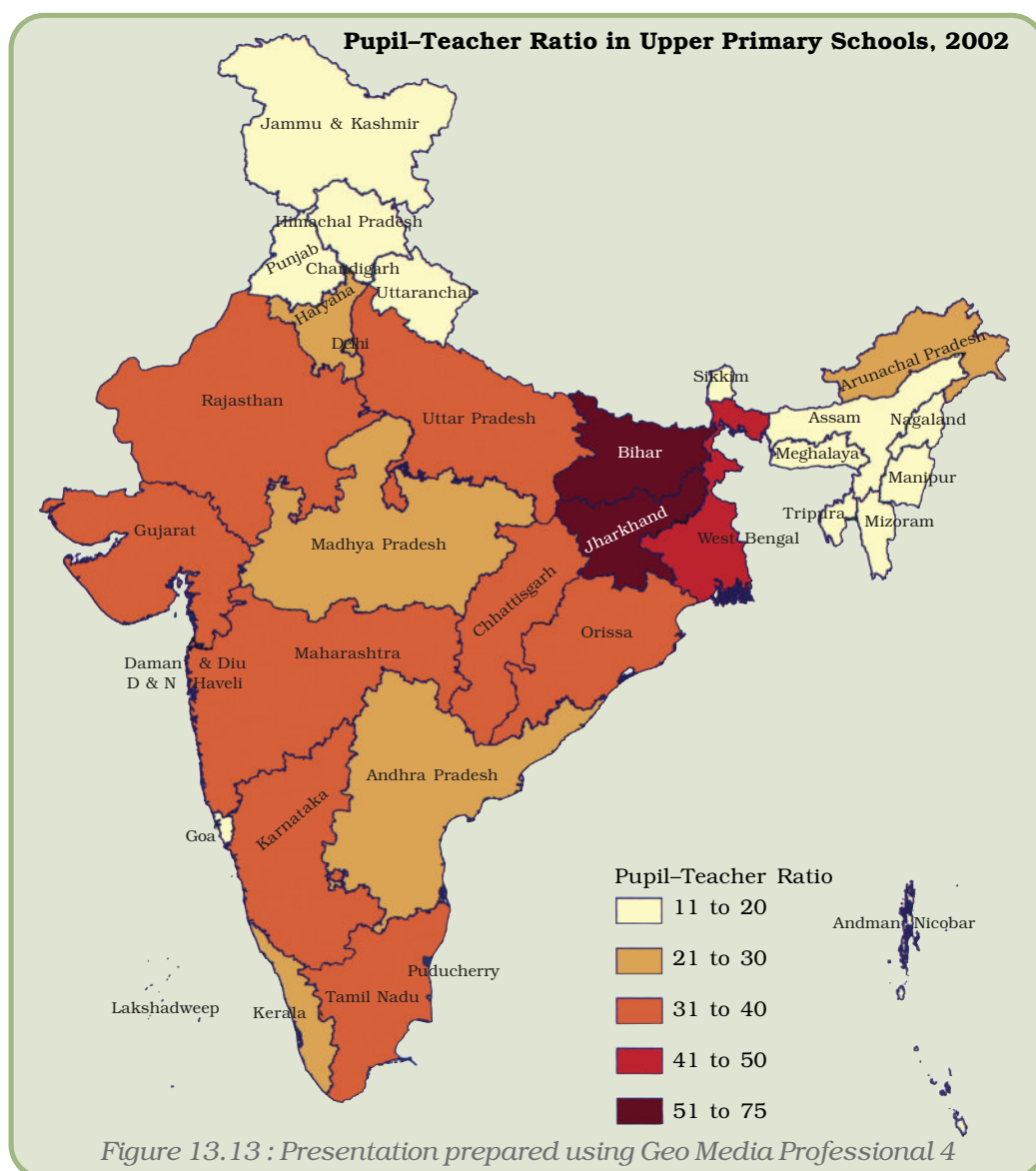
Source : VII All India School Education Survey

13.6 NANOTECHNOLOGY

Nanotechnology is a technology for building or constructing materials, devices, tools etc. in smallest or lowest possible form i.e. on the scale of atom and molecule. One nanometre is a one billionth of a meter.

Nanotechnological inventions would bring enormous benefits and luxury in human life. It is widely believed that nanotechnology has got the much needed potential to be effective in terms of energy consumption besides being environment friendly. It is also expected to solve major health problems. Applications of this technology will help manufacturing products at reduced cost which will be smaller, lighter and cheaper.

The Nanotechnology is believed to be very promising in bringing solutions in the fields of health and sanitation, food security and environmental issues.

Summary

- Improvements in Hard disk drive performance have resulted from faster data rates, faster seek times, larger cache sizes, higher interface speeds, advances in microcode algorithms and most importantly higher spindle speed.

- Emerging trends in Hard Disk Drives include Enterprise HDDs, Hybrid HDDs, Full Disk Encryption (FDE) and Micro Electro Mechanical Systems (MEMS).

- Enterprise HDDs are specifically designed for mission-critical applications such as core servers and large-scale storage systems.

- A Hybrid Hard Disk Drive combines a conventional HDD and non-volatile flash memory.

- Full Disk Encryption (FDE) is a technology where data is encrypted before storage.

- MEMS-based storage has attractive features such as small size, shock resistance, and low-power consumption.

- Redundant Array of Inexpensive Disks (RAID) is used to store the same data at different places redundantly on multiple hard disks.

- Network storage like SAN and NAS allows consolidation of data spread over various parts of the organisation and reduces duplication of information.

- Online storage options provide viable storage solutions through normal Internet connections for small size companies.

- Using 3D optical data storage technology, data is stored in multiple layers in the optical disc.

- Holographic storage system is an emerging technology where data is stored through full depth of the media instead of only on the surface.

- Universal Serial Bus (USB) has become the technology of choice for connecting peripherals to PCs, cell phones, printers etc.

- Wireless USB allows connectivity to low speed devices at reasonable cost.

- Architecture of the microprocessor has changed from single processor to embedded computing devices where multiple microchips and processors are combined to form a package to meet system requirements.

- Research for developing new materials to design semiconductors which are compact, fast and energy efficient are going on.

- Optical communication system makes use of devices for maximum utilisation of the Optical backbone.

- Development in the field of wireless technology has led to more reliable, higher bandwidth and spectrum efficient devices.

- Open Source software are freely available and can be edited and customised by anyone.

- Software development has become more systematic with great emphasis on requirement engineering, prototyping and testing.

- An important trend in software is the use of component driven software development where software developers can select components from comprehensive libraries of reliable and well-documented software and combine them together to carry out the required purpose without creating everything from scratch again and again.

- Computer-aided design (CAD) programs are used to create two and three-dimensional architectural drawings, engineering drawings, and product designs. Computer Aided Manufacturing (CAM) systems use CAD to help manufacture the products.

- Mapping software combines capabilities of graphics, database management, and spreadsheet to display the data geographically which gives better understanding of geographical patterns behind the data.

- With Photo image editors you can manipulate and revise photos and other images that you have scanned with an optical scanner.

- Nanotechnology is the new field for developing material, devices, tools etc. in the smallest possible form on the scale of atoms and molecules.

EXERCISES

SHORT ANSWER TYPE QUESTIONS

1. Give any two examples of dual core processors.

2. Describe the characteristics of Xeon processors.

3. Differentiate between 3-D optical and holographic memories.

4. What is Nanotechnology?

5. What is component driven software development?

6. How Graphene is expected to improve the computer processing speed?

LONG ANSWER TYPE QUESTIONS

1. What is a processor? Describe different types of processors.

2. What are the new technologies used in processors?

3. Which different physical characteristics do a MEMS-based storage has as compared to a traditional disk?

4. Describe how binary data is stored and represented in 3-D optical storage devices.

5. What is network storage? Describe various upcoming network storage trends.

6. Mention the roles software play in our daily life. Describe with examples.

7. Describe the advantages of using a mapping software.

8. Compare educational software and computer game software.

9. Categorise different application software according to their use and give examples for each.

Activities

1. Make a comparative chart of different types of processors.

2. Find out and compare the data transfer speeds of the various data storage devices that you use like hard disks, CDs, DVDs, flash disks and others.

3. Take any unusable/malfunctioning CD and DVD and study their cross-sectional view. Observe and compare the functionalities of the different layers contained in them.

4. For each of the emerging data storage device that you have studied find out the status of its commercial production the company and the device that has been launched or is yet to be launched.

5. Find out if some organisation or institute that is accessible to you uses a network storage setup and study it.

REFERENCES

Books

Emerging Web Services Technology by Cesare Pautasso and Christoph Bussler, Publisher: Birkhause Verlag, P.O. Box 133, CH -4010 Basel, Switzerland

URLs
1. www.wikipedia.org
2. www.intel.com
3. www.amd.com
4. www.cyrix.com
5. www.research.ibm.com
6. www.ieeexplore.ieee.org
7. www.articlesbase.com
8. www.science.nasa.gov
9. www.experts-exchange.com
10. www.ietcom.oxfordjournals.org
11. www.crnano.org
12. www.webopedia.com

COMPUTER CONTROLLED DEVICES

14

Objectives

After completing this Chapter, the student will be able to:

- *develop understanding about computer controlled devices and sensor controlled devices,*

- *describe uses of systems or devices that are controlled by computers,*

- *cite examples of where computers control system in response to physical input data captured through sensors and*

- *take the learning from the chapter and use it to apply in understanding computer controlled devices around him/her.*

A machine may be deemed intelligent when it can pass for a human being in a blind test.

-Alan Turing

Introduction

Technology has touched almost all aspects of our life. Whether it is banking, telecom, communication, health, engineering or automated processes computer controlled devices not only make our lives convenient but also enable us to take important decisions.

So far we have been learning about various usage of computers and its functioning and also its varied uses in our daily lives. In the present chapter we will learn more about devices that work by computer controlled mechanisms. These devices are termed as computer controlled devices that have become part of our daily living in many ways; some of them are simple while on the other hand some are very complex in their functions.

This chapter would help students to familiarise with applications of such devices, and sensors that work in conjunction with computers. First let us understand, what is the meaning of a computer controlled device? "A computer controlled device is a device that accepts physical inputs from external source and pairs this information with computer controlled instructions to perform specific functions."

These devices are not necessarily desktop computers that we are familiar with, but are embedded in other systems, controlling and coordinating their functioning.

Let us see some examples of computer controlled devices that we come across in our day to day lives. We all have been familiar with robots, which are able to perform a host of tasks; this is one of the most common example of such a device. It is programmed to move, manipulate objects, and accomplish work while interacting with its environment.

14.1 SENSORS

Some popular forms of computer controlled devices that we readily encounter are sensors of a wide variety. A sensor is defined as "a device that receives signal or stimulus (as heat, pressure, light etc.) and responds to it in a distinctive manner". These sensors are used at railway stations in escalators, in weighing machines, and also in robots that have specific functions.

Interestingly it is to be noted that the way robots function is not the same as the way sensor operated devices function. However sensors are also used extensively as embedded component in robots that have to perform complex functions like surgery or detection of explosive in public places.

Robots mostly work on a pre-installed instruction program that has a control through a computer assembly whereas sensors function on the inputs that they receive in physical form from the external environment and then they convert this input into electronic signals and that finally leads to a desired action or response. In the following part of this chapter we would see the usage of sensors as computer controlled devices in and around us.

There is an enormous variety of devices in which the computer functions on the basis of the input provided to it through a wide variety of devices called sensors. A sensor is a device that senses and measures physical attributes like temperature, pressure etc. and converts it into a signal which is read by an observer or by an instrument particularly.

The commonly used barcode is perhaps the most common example of how a sensor collects data and makes it computer readable. Barcodes are familiar black and white stripes on packaged goods, containing information on the type of an item, identity of its manufacturer, its price,
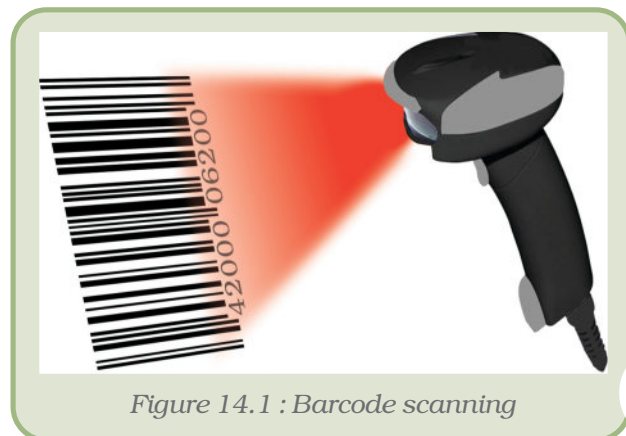

Figure 14.1 : Barcode scanning

377

etc. A barcode is read by shining light at it. A barcode scanner, which is an optical device, reads the barcode. It requires a line-of-sight condition; there should not be any obstruction between the barcode and the beam of light.

Parts of the code reflect the light and parts do not. The barcode communicates its fixed information by spatially modulating the incident light. The communication is read by a computer and displayed in a language understood by all.

14.1.1 OTHER EXAMPLES OF SENSORS

Motion Detection

Motion of an object is detected electronically using infrared light or laser technology and acoustic devices. Motion detectors have sensors that detect movement and send a signal to a sound device that produces a sound alarm. There are motion detectors that employ cameras to capture images and send them to a computer. Such sensors are used in banks, shopping malls etc.

Parking Sensors (Based on Distance Sensors)

Parking sensors are a technology that allows the driver of an automobile, truck, or commercial vehicle to be alerted to nearby objects in their path. Parking sensors are also known as backup sensors, parking sonar or just sonar depending on the automaker. These sensors send ultrasonic waves 40,000 times per second that detect obstacles behind the vehicle. A controller installed inside the vehicle then receives the signal from the sensors and transmits it to a speaker that emits a tone, or to a combination of speaker and display unit.

A smart sensor collects data and converts it into result in a unit appropriate for the particular physical attribute that is being measured. A smart sensor can monitor voltage, radiation, temperature, humidity, etc. and process the information by itself and adjust the system.

Location Tracking (GPS)

A wide variety of amazingly powerful antennas are used to locate objects of various kinds that are often dynamic and change their position with time.

The Global Positioning System (GPS) tells us where we are on earth by generating absolute location data. GPS satellites orbit the earth twice a day and transmit signals to GPS receivers on earth. These receivers are equipped with smart sensors to capture the signals and calculate the exact location of an object.

Once the position of the object is established, the GPS unit can calculate other information, like speed at which it is moving, the track it follows, the places it has been through during the trip, its distance to destination, and more.
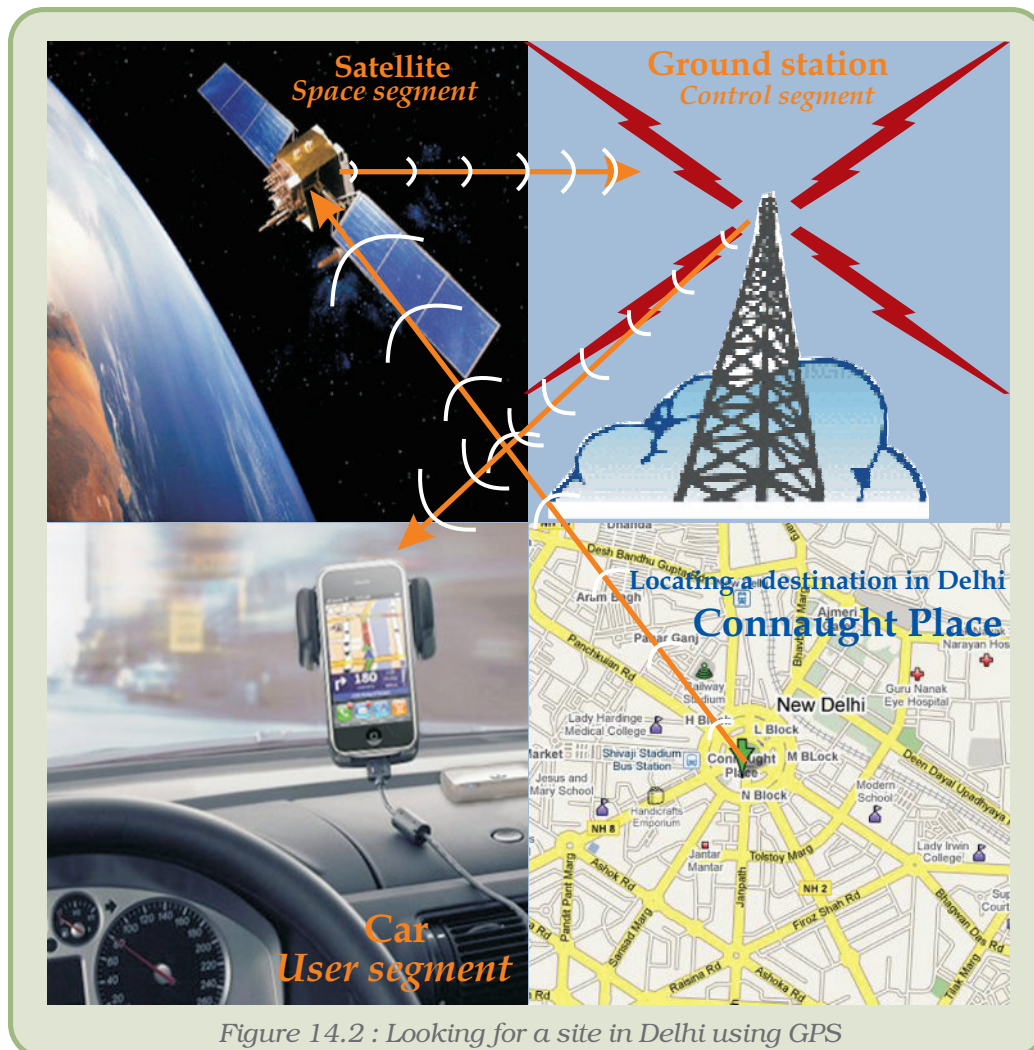
Figure 14.2 : Looking for a site in Delhi using GPS

GPS has three segments. The **space segment** consists of 28 satellites that are 11,000 nautical miles above the earth. The **user segment** consists of receivers, which one can hold in ones hand or mount in the car. The third one, the **control segment**, consists of five ground stations around the world.

GPS devices are installed in cars and taxis and in all aircrafts.

14.2 AUTOMATED TELLER MACHINE (ATM)

So far we had seen various types of robots and sensor devices. Now we would also try to understand another very common example of computer controlled device that

Figure 14.3 : Car mounted GPS device for local navigation

379

we keep using frequently in our daily lives, the ATM. From the ATM teller we get access to a host of services like bank account related information, cash deposit, cash withdrawal etc. try to understand how this computer controlled device functions in a slightly greater detail.

An ATM | Block diagram of ATM

*Figure 14.4 : An ATM teller machine*
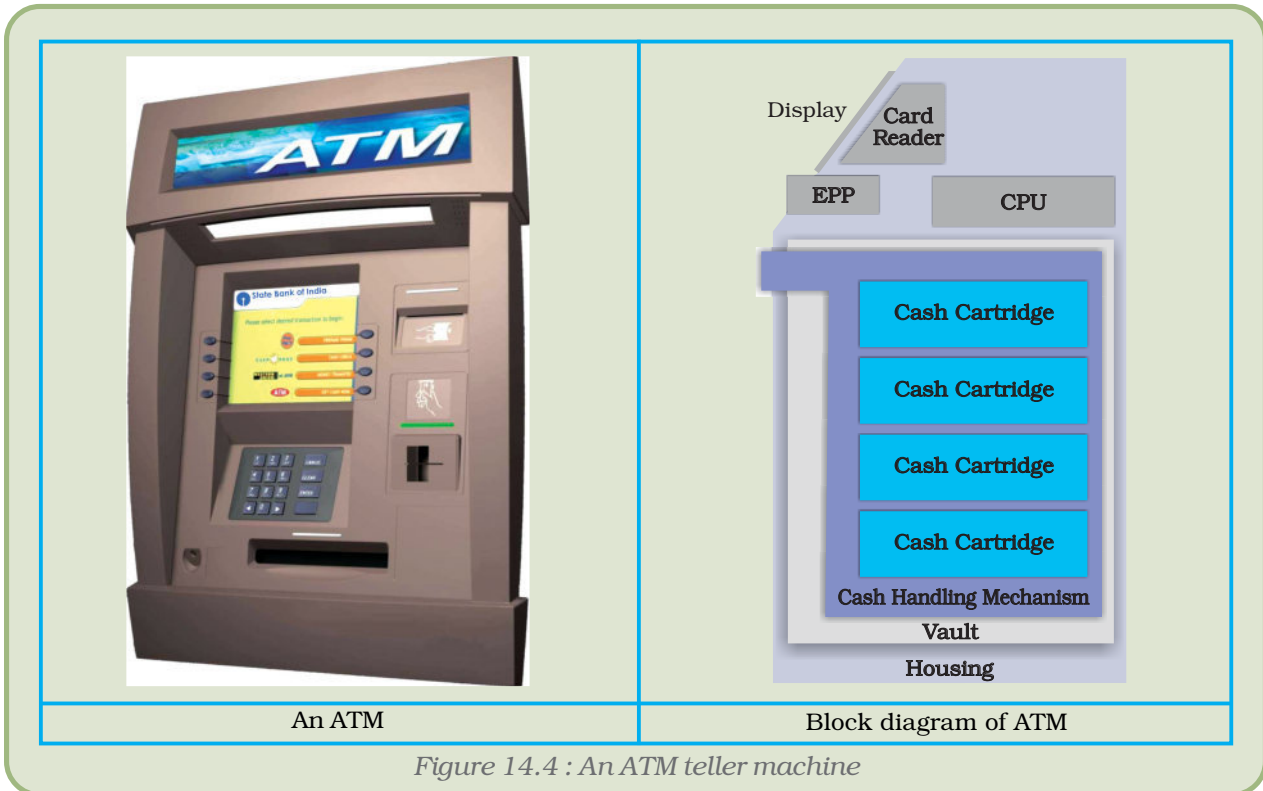
An ATM is a computer controlled machine designed to dispense cash, and other bank information without going to the bank. The ATM can also take deposits, transfer money between bank accounts along with basic banking services. This electronic machine is connected to the bank database through a computer controlled system that has capability to perform functions based upon the information that is provided by the customer through its banking card, it can be a debit card or credit card.

The magnetic tape on the card is read by the ATM machine that checks the credentials of the user, once it is verified, a list of services is visible on the screen of the ATM. The customer as per his requirement can choose one option at a time, further this request of customer is taken by the machine and it then processes it and delivers the service requested by the customer.

Here interestingly we can see how effectively computer controlled devices can be synchronised between the service provider (that is the bank in this case) and the customer and make the entire banking service very much convenient and can be utilised by the customer round the clock suiting his/her convenience.

The following figures 14.4 and 14.5 display an ATM, the block diagram and a systems overview of an ATM which is computer controlled and organsation of different units and their workings through a computer instucted program mechanism.

Figure 14.5 : A systems overview of an ATM
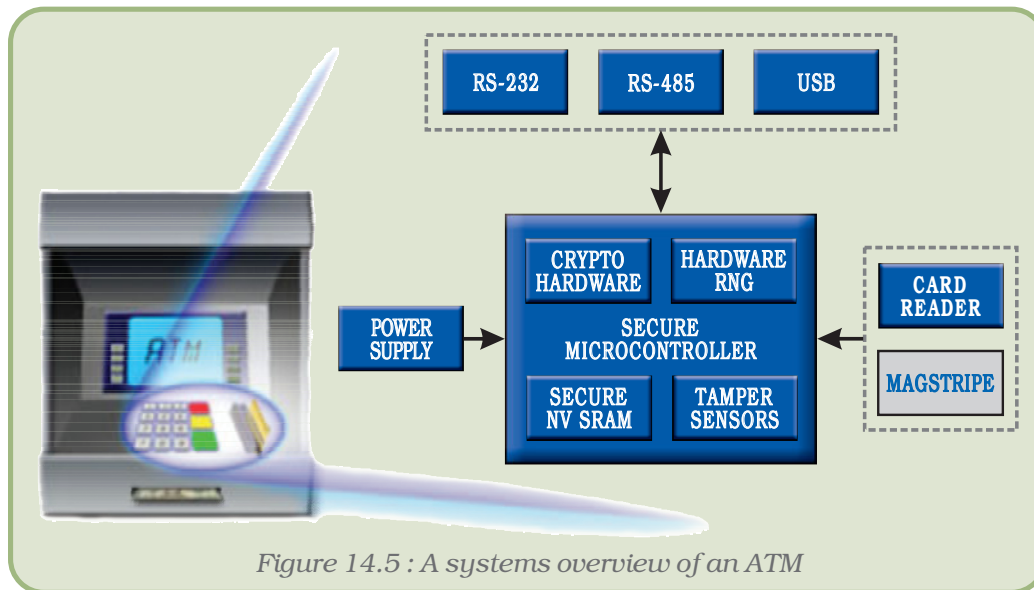
14.3 ROBOT

A robot is defined as a computerised system with a motorised construction (usually an arm) capable of interacting with the environment. In its most basic form, it contains sensors, which provide feedback data on the robot's current situation, and a system to process this information so that the next action can be determined.

Robots are able to perform repetitive tasks quickly, cost effectively, and accurately than humans. This includes robotic arms and motors (also known as actuators). Most advanced robot arms make use of sensors like motion and pressure sensors in order to detect foreign obstacles and avoid breaking or dropping what it is carrying.

We see applications of robots to perform a host of tasks ranging from simple picking up of objects to assembling of parts. For example in the automobile industry, a customised program is preinstalled in the robots through a computer that controls the functioning of the robot in the desired manner. The same robot can perform diversified functions depending upon the type of programs that are installed into it.

Modern banking would not be possible without these devices and on the versatile applications of computers. The Automated Teller Machines (ATM) are a familiar example of everyday application of computer controlled devices. If you want to withdraw cash from an ATM, you insert your ATM card in the machine. It verifies your credentials, takes up the request, processes it, and then dispenses the cash amount along with a brief statement.

The list of such systems in use today is indeed very long. The examples chosen here are those of familiar applications of sensors of various kinds, controlled/coordinated and monitored by computers.

14.3.1 VARIOUS USES OF ROBOTS

The automobile industry makes use of robots that perform a host of complex and repetitive functions with precision, accuracy and speed. They are computer controlled devices where a specific program
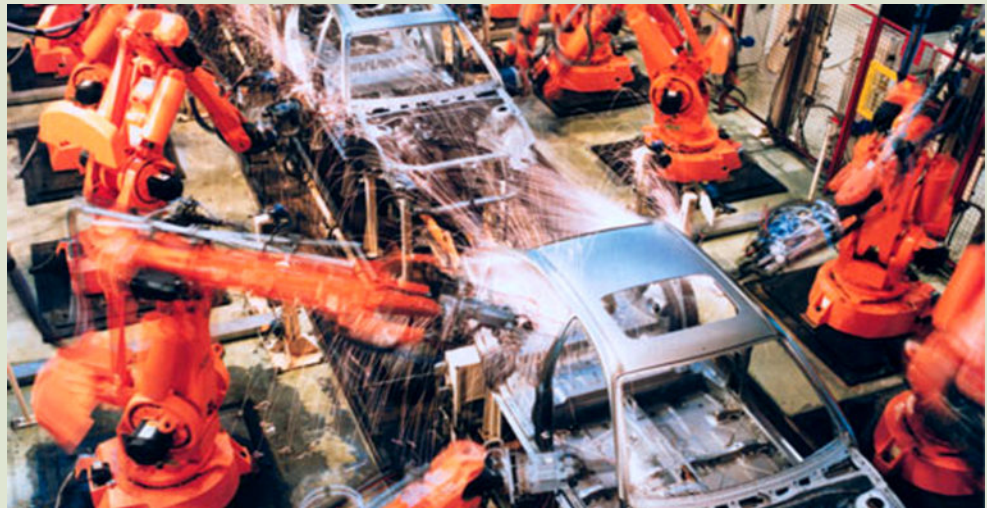

Figure 14.6 : Robot welding the car body

instruction is installed and then it is executed through a computer. Whether it is to assemble spares or to put the fiber sheets into specific moulds as a car, robots function effectively with great amount of efficiency. The usage of robots has revolutionised the automobile industry.
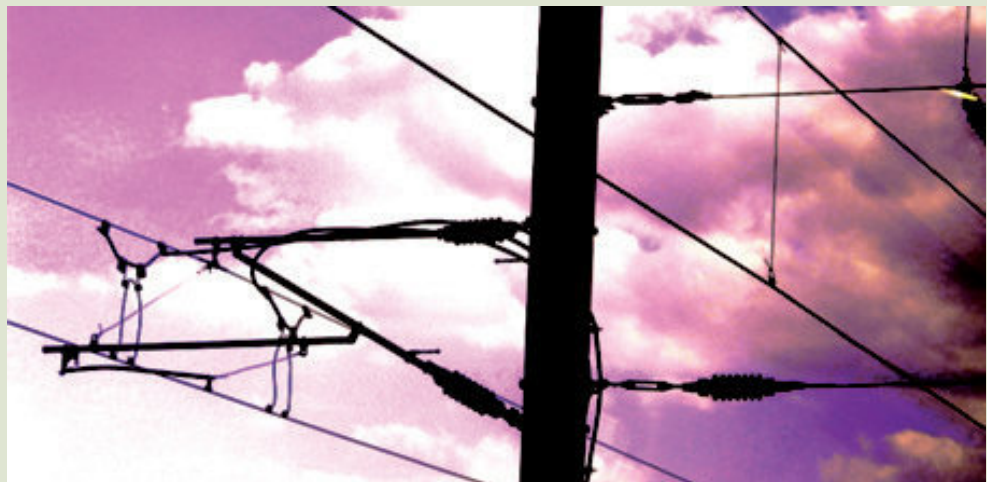

Figure 14.7 : Electric connection system using robot arm in an industrial setting

Large scale infrastructure industries especially in high risk prone areas use electric arm robots to manage the electrical connection system for power distribution. These robots are computer controlled to perform such high risk tasks with precision and accuracy.

Figure 14.8 : Robot picking up inert unexploded material from ground

Security agencies also adopt such devices to locate explosives and unexploded material at public places like airports, industries railways, roadways and stations to name a few. These robots not only are able to detect the explosives beforehand but in some cases they can be also be used to de-circuit an explosive assembly without any human interface. This helps in reducing casualties of officers on duty in security agencies considerably.

Figure 14.9 : A robotic operation theatre

Robotic surgery, on the other hand, requires the use of a surgical robot, which may or may not involve the direct role of a surgeon during the procedure. One key advantage of robotic surgery over computer-assisted is its accuracy and ability to repeat identical motions.

It is essential to understand that robots by virtue of performing a host of simple to complex functions, are designed from simple computer controlled ones like assembly line robotic arms in production units, to more complex ones that have sensors and actuators installed in them like the surgical robots that can perform complex surgeries without the physical involvement of the surgeon.

So far, through various examples we have seen that how significant computer controlled devices have become in our lives. These devices are complex and developed after a lot of research that involves integration of artificial intelligence and a pre-program instruction that determines what the robot would do and controlled by a distant computer.

Summary

- A computer controlled device accepts physical inputs from external source and pairs this information with computer controlled instructions to perform specific function.

- Most of the computer controlled devices make use of sensors.

- A sensor is a device that receives signal or stimulus (e.g. voltage, heat, pressure, radiation, sound waves, humidity etc.) and responds to it in a distinctive manner.

- The sensors find its utility at variety of machines/devices like bar code scanners, weighing machines, escalators, automatic flush, ATM, robots.etc.

- The barcode scanner is an optical device, which reads the barcodes by shining light over these. The barcodes (familiar black and white stripes on packaged goods) contain information on the identity of item, its manufacturer, its price, etc.

- A smart sensor collects data and process the information into result in a unit appropriate for the particular physical attribute.

- A robot is a computer controlled device programmed to move, manipulate objects, and accomplish work while interacting with its environment.

- The robot, in its most basic form, contains sensors, which provide feedback data on the robot's current situation, and a system to process this information according to pre-program instructions so that the next action can be determined accordingly.

- The robots perform a host of complex and repetitive jobs precisely, accurately and quickly. A few of the jobs performed by robots are:
  ◦ car welding,
  ◦ surgery,
  ◦ detecting and picking foreign obstacles without breaking or dropping what it is carrying,
  ◦ locating explosives and de-circuiting explosive assemblies saving risks to human beings and
  ◦ managing electrical connection systems in high risk zones for power distribution etc.

- An Automated Teller Machine (ATM) is a computer controlled machine designed for financial transactions without going to the bank.

- The ATM is connected to the bank database through a computer controlled system that has capability to perform transactions based upon the information that is provided by the customer through its banking card (debit/credit card).

EXERCISES

SHORT ANSWER TYPE QUESTIONS

1. How does a sensor work? Describe its usages at various places.

2. What are different sectors where robots find their usage? Discuss your response with at least three examples.

3. What are some of the limitations of computer controlled devices? By looking at the examples derive the probable limitations of computer controlled devices.

4. "Weather monitoring through GPS is an example of computer controlled device." Analyse this statement and present your views either for or against the statement.

5. Explain the functioning of an Automated Teller Machine (ATM) and how it makes the banking services more convenient to both the service provider and the customer.

Activities

For class discussion :

1. Form a study group of five to seven students per group. Discuss the usage of computer controlled devices and prepare a list of important areas where they are used.

2. List out some of the important applications of such devices.

3. "What is the difference between a sensor and a computer controlled device?". Discuss in small groups of five to eight students.

4. You are traveling from Delhi to Kolkata by train. As you reach Allahabad railway station, halfway between Delhi and Kolkata, your mobile receives a message saying that you are at Allahabad. How does this happen? Is it GPS that works?

5. Choose one computer controlled device and prepare a presentation explaining the following :
   a) What is the need of such a device?
   b) What are its key components and in simple terms how does it function?
   c) What are other areas in our environment that can be assisted with computer controlled devices?