

**Vocational Higher Secondary
Education (VHSE)**

Second Year

**COMPUTER SCIENCE &
INFORMATION TECHNOLOGY**
Reference Book



Government of Kerala
Department of Education

State Council of Educational Research and Training (SCERT),
KERALA
2016

List of Contributors

Participants

1. **SHYJITH B. T.**
Vocational Teacher in CSIT
TKDM GVHSS Kadappakada, Kollam
2. **SREERAJ P.**
Vocational Teacher in CSIT,
V.V.H.S.S, Poredom, Kollam
3. **ASHA NAIR S.**
Vocational Teacher in CSIT
G.V.H.S.S, Mavelikkara, Alappuzha
4. **SYAM D.**
Vocational Instructor in CSIT
K.P.S.P.M.V.H.S.S, East Kallada, Kollam.

Experts

1. **Dr. K Sanil Kumar**
Associate Professor & Head, Dept. of Computer Science,
SreeAyyappa College, Eramalikkara, Chengannur
2. **Shri. Neevan R**
Assistant Professor, Dept. of Computer Science,
College of Engineering, Attingal
3. **Viji Balakrishnan**
Asst. Professor in Computer Science & Member,
Board of Studies, PG Computer Science, University of Kerala,
College of Applied Science (IHRD), Mavelikkara
4. **Anil Kumar**
Draftman (Mechanical)
Govt. Polytechnique, Ezukone, Kollam

Academic Co-ordinator

Dr. Gopalakrishnan N
Research Officer, SCERT

Prepared by :

State Council of Educational Research and Training (SCERT)

Poojappura, Thiruvananthapuram 695012, Kerala

Website : www.scertkerala.gov.in e-mail : scertkerala@gmail.com

Phone : 0471 - 2341883, Fax : 0471 - 2341869

Typesetting and Layout : SCERT

© Department of Education, Government of Kerala

Foreword

Dear Learners,

This book is intended to serve as a ready reference for learners of vocational higher secondary schools. It offers suggested guidelines for the transaction of the concepts highlighted in the course content. It is expected that the learners achieve significant learning outcomes at the end of the course as envisaged in the curriculum if it is followed properly.

In the context of the Right- based approach, quality education has to be ensured for all learners. The learner community of Vocational Higher Secondary Education in Kerala should be empowered by providing them with the best education that strengthens their competences to become innovative entrepreneurs who contribute to the knowledge society. The change of course names, modular approach adopted for the organisation of course content, work-based pedagogy and the outcome focused assessment approach paved the way for achieving the vision of Vocational Higher Secondary Education in Kerala. The revised curriculum helps to equip the learners with multiple skills matching technological advancements and to produce skilled workforce for meeting the demands of the emerging industries and service sectors with national and global orientation. The revised curriculum attempts to enhance knowledge, skills and attitudes by giving higher priority and space for the learners to make discussions in small groups, and activities requiring hands-on experience.

The SCERT appreciates the hard work and sincere co-operation of the contributors of this book that includes subject experts, industrialists and the teachers of Vocational Higher Secondary Schools. The development of this reference book has been a joint venture of the State Council of Educational Research and Training (SCERT) and the Directorate of Vocational Higher Secondary Education.

The SCERT welcomes constructive criticism and creative suggestions for the improvement of the book.

With regards,

Dr. P. A. Fathima
Director
SCERT, Kerala

CONTENTS

<i>ABOUT THE COURSE</i>	5
<i>SYLLABUS</i>	6
<i>FUNDAMENTALS OF COMPUTER LANGUAGES</i>	10
<i>PROGRAMMING IN C++</i>	17
<i>ARRAYS AND STRINGS</i>	27
<i>STRUCTURE</i>	31
<i>FUNCTIONS</i>	34
<i>OBJECT ORIENTED PROGRAMMING</i>	39
<i>POLYMORPHISM AND INHERITANCE</i>	48
<i>FILES IN C++</i>	53
<i>DATABASE DESIGNING</i>	56
<i>SQL (STRUCTURED QUERY LANGUAGE)</i>	65
<i>ENGINEERING GRAPHICS</i>	77
<i>INTERNET AND CYBER SECURITY</i>	95
<i>WEB DESIGN USING HTML</i>	100
<i>ADVANCED HTML</i>	112
<i>CLIENT SIDE SCRIPTING</i>	125
<i>SEVER SIDE SCRIPTING</i>	140
<i>WEB HOSTING</i>	160
<i>TRENDS IN COMPUTING TECHNOLOGY</i>	165
<i>LIST OF REFERENCE</i>	176

ABOUT THE COURSE

The COMPUTER SCIENCE AND INFORMATION TECHNOLOGY (CSIT) aims to equip students the knowledge, skills and attitudes to become productive employees in the area of computer science and technology. In this course, students will gain a perspective to become a successful entrepreneur in Computer Science & Information technology and allied sector.

Scope of the Course

The Computer Science and Information Technology course also provide students with the foundation for higher studies. Computer science and Information technology is one of the most significant growth catalysts for the Indian economy. Computer science and Information technology not only influenced the employment prospects of the people but also affected the social lives of the people through networking and social web sites. CSIT course certificate is approved for various posts notified by Kerala Public Service Commission (KPSC).

Course Nature

The main objective of Vocational Higher Secondary Education is to make a self reliance society who are capable of being self employed, acquiring skills to get job as well as to obtain higher studies. In tune with the modular approach, the CSIT course consists of 4 modules. Two or more job roles are identified in each module. Upon the completion of each module, skill certificate is awarded to students.

Main Focus Areas

- Hardware Installation and maintenance
- Network configuration
- DTP with Malayalam
- Computer Programming
- Computing Techniques
- Web Development
- Laptop maintenance
- CCTV configuration
- Graphics design
- Database Designing
- Cyber Ethics

Relevance of the Course

Many job opportunities in CSIT are emerged across the world. Upon the completion of the course, students will acquire skill and competent to enter in the upcoming job market. Much of the CSIT related activities are entered on service in IT Field, Banking, Tele communication, Networking, Programming; Web designing, etc.

SYLLABUS

Module 3: Object Oriented Programming and Databases

Unit No 3.1- Fundamentals of Computer Languages

Periods: 30

- 3.1.1 Types of computer languages
- 3.1.2 Language translators
- 3.1.3 Approaches in problem solving
- 3.1.4 Programming methodologies
- 3.1.5 Algorithm and flowchart

Unit No 3.2 - Programming In C++

Periods: 140

3.2.1. Introduction to C++

- 3.2.1.1 Character set and tokens
 - o Input/Output statements
 - o Turbo and Geany development environment
- 3.2.1.2 Data types
- 3.2.1.3 Control statements
 - o Branching statements
 - o Looping statements
 - o Jump statements

3.2.2 Arrays and Strings

- 3.2.2.1 Arrays
- 3.2.2.2 Strings
- 3.2.2.3 String manipulation functions

3.2.3 Structure

- 3.2.3.1 Structure
- 3.2.3.3 Nested structure
- 3.2.3.4 Array of structure

3.2.4 Functions and Pointers

- 3.2.4.1 Built-in functions
- 3.2.4.2 User defined functions
- 3.2.4.3 Parameter passing
- 3.2.4.4 Pointers

3.2.5 Object Oriented Programming

- 3.2.5.1 Object Oriented Programming Concepts
- 3.2.5.2 Class and Objects

- 3.2.5.3 Member functions
- 3.2.5.4 Friend functions and Friend class
- 3.2.5.5 Constructors and Destructors

3.2.6 Polymorphism and Inheritance

- 3.2.6.1 Polymorphism
- 3.2.6.2 Implementation of Polymorphism
 - o Function overloading
 - o Operator overloading
- 3.2.6.3 Inheritance

3.2.7 Files in C++

- 3.3.7.1 File stream classes
- 3.3.7.2 File modes
- 3.3.7.3 File operations
 - o Read ()
 - o Write ()
- 3.3.7.4 Error handling functions

Unit No 3.3 - Database Designing

Periods: 40

- 3.3.1 Concept of DBMS
- 3.3.2 Advantages of the database management system
- 3.3.3 Data abstraction and Data independence
- 3.3.4 Data models
- 3.3.5 Terminologies in RDBMS.
- 3.3.6 Keys in RDBMS

Unit No 3.4 - SQL (Structured Query Language)

Periods: 80

- 3.4.1 Features of Structured Query Language.
- 3.4.2 Data types and commands
- 3.4.3 Practice on SQL commands
- 3.4.4 Clauses associated with DML commands
- 3.4.5 Aggregate Functions

Unit No 3.5- Engineering Graphics

Periods: 50

- 3.5.1 Application areas of Engineering Graphics
- 3.5.2 Drawing Instruments
- 3.5.3 Lettering, Numbering and Dimensioning.
- 3.5.4 Lines

- 3.5.5 Scales
- 3.5.6 Geometric construction
- 3.5.7 Basic geometric shapes
- 3.5.8 Projection
- 3.5.9 Orthographic and isometric projection

Module 4 Web Application Development

Unit No 4.1 - Internet and Cyber Security

Periods: 45

- 4.1.1 History of Internet
- 4.1.2 Terminologies
- 4.1.3 Cyber ethics
- 4.1.4 Legal Issues

Unit No 4.2 - Web Design Using HTML

Periods: 45

4.2.1 BASIC HTML

- 4.2.1.1 Web pages
- 4.2.1.2 Static and Dynamic web pages
- 4.2.1.3 Tags in HTML
- 4.2.1.4 Structure tags
- 4.2.1.5 Formatting tags
- 4.2.1.6 Comments in an HTML document
- 4.2.1.7 Image Tags
- 4.2.1.8 Linking tag
- 4.2.1.9 List Tags

4.2.2 - Advanced HTML

Periods: 55

- 4.2.2.1 Table Tags
- 4.2.2.2 Form Tags
- 4.2.2.3 Frame tag
- 4.2.2.4 Cascading style sheet
- 4.2.2.5 Multimedia contents

Unit No 4.3 - Web Development using Javascript & PHP **Periods: 50**

4.3.1 CLIENT SIDE SCRIPTING

- 4.3.1.1 Scripting languages
- 4.3.1.2 Importance of JavaScript
- 4.3.1.3 Data Types & Variables

- 4.3.1.4 Operators
- 4.3.1.5 Control Structures
- 4.3.1.6 Functions
- 4.3.1.7 Accessing values through a text box

4.3.2 Server Side Scripting

Periods: 100

- 4.3.2.1 Features of PHP
- 4.3.2.2 Syntax of PHP statement
- 4.3.2.3 Comments
- 4.3.2.4 Variables and Constant
- 4.3.2.5 Data types and Operators
- 4.3.2.6 Control structures
- 4.3.2.7 Functions
- 4.3.2.8 Arrays
- 4.3.2.9 PHP Forms
- 4.3.2.10 PHP global variables and Super Global arrays
- 4.3.2.11 Database functions used in PHP
- 4.3.2.12 Database connectivity

Unit No 4.4 - Web Hosting

Periods: 15

- 4.4.1 Types of web hosting
- 4.4.2 Buying hosting space
- 4.4.3 Domain name registration
- 4.4.4 FTP software

Unit No 4.5 - Trends in Computing Technologies

Periods: 30

- 4.5.1 Mobile network
- 4.5.2 Generation of networks
- 4.5.3 Wireless networking technologies
- 4.5.4 Mobile application development
- 4.5.5 Computing technologies
- 4.5.6 IoT
- 4.5.7 Artificial Intelligence

MODULE 3**OBJECT ORIENTED
PROGRAMMING AND DATABASES****Unit - 1****Fundamentals of Computer Languages**

This unit introduces how computer is used for problem solving. The first part describes the different types of programming languages and translators. Here also discuss various approaches (Top Down and Bottom Up) in problem solving. Upon completion of problem solving approaches, introduces algorithm and flow chart A computer can solve problems only when we give instructions to it. An instruction is an action oriented statement.

Learning Outcomes

The learner :

- Classifies different types of computer languages
- Identifies different language translators
- Differentiates Top Down and Bottom Up approaches in problem solving
- Describes different programming methodologies
- Develops algorithm and flowchart for various problems

3.1.1 Types of Computer Language

Just as humans use language to communicate, and different regions have different languages, computers also have their own languages that are specific to them. Different kinds of languages have been developed to perform different types of work on the computer. Basically, languages can be divided into two categories according to how the computer understands them.

Two Basic Types of Computer Language

- Low-Level Languages:
- High-Level Languages:

Low-level Languages

A language that corresponds directly to a specific machine. Low-level computer languages are either machine codes or are very close them. A computer can only understand and execute instructions given in the form of machine language i.e. binary.

There are two types of low-level languages:

- Machine Language
- Assembly Language

Machine Language

Machine language is a language that is directly interpreted by the hardware. It is the lowest level of programming language, which uses binary codes. ('1' and '0').

Assembly Language

Assembly language is slightly more user-friendly language that directly corresponds to machine language. Assembly language use mnemonic codes. It was developed to overcome some of the many inconveniences of machine language.

High-level Languages

High-level computer languages use formats that are similar to English. It is very easy to use and understand. High-level languages are basically symbolic languages that use English words and mathematical symbols.

Examples of High-level Languages

C++ Language: The C++ language has an object oriented structure. The structure of object oriented also permit code to be reused many times. This language is an efficient language.

Java Language: The Java language is a multi-platform language that's particularly helpful in networking. It is similar to C++ in structure and syntax. Java is very easy to learn.

PHP Language: The PHP language is used to design dynamic web pages.

3.1.2 Language Translator

All the programs written in programming language (except machine level language) is first translated into machine understandable form for its successful execution. Assemblers, Compilers and Interpreter are the examples of Language translators.

Assembler: An assembler translates assembly language into machine code.

Compiler: A Compiler translates code written in a high level language to a lower level language, object/machine code.

Eg: Computer languages that use compiler : C, C++...

Source Program → Compiler → Target Program

Interpreter: An interpreter executes the programs directly, running through program code and executing it line-by-line.

Eg : Computer Languages that used interpreter : BASIC

Difference between Compiler and Interpreter

Both compilers and interpreters are used to process high level programming languages. But they are quite different in features.

Compiler	Interpreter
It translates a whole program at a time	It translates a line of code at a time
It generates intermediate codes after translation	No intermediate codes generated
Errors are detected only after compilation	Errors are known at each line
More memory space required (since intermediate code is generated)	Less memory space required
Compiled programs are faster	Interpreted programs are rather slow
Can be used for both small and huge size program	Only used in small blocks of code.

3.1.3 Approaches in Problem Solving

1. Top down design (top down decomposition)
It is the process of breaking the overall procedure or task into modules and then subdividing each module until the lowest level of detail is reached.
2. Bottom up design
The solution for the main module will be developed only after designing specific solution to its sub modules.

3.1.4 Programming Methodologies

1. Procedural programming:-Important features of procedural programming
 - a. Programs are organized in the form of subroutines
 - b. All data items are global
 - c. Program controls are through jumps and calls to subroutine
 - d. Subroutines are abstracted to avoid repetitions
 - e. Suitable for small sized software application
 - f. Difficult to maintain and enhance the program code
Eg:- FORTRAN and COBOL
2. Structured programming:-Programs consists of multiple modules, each module has a set of functions of related types. Important features of structured programming
 - a. Emphasis on algorithm rather than data

- b. Programs are divided into individual procedures
- c. Procedures are independent of each other
- d. Procedures have their own local data
- e. Introduction of the concepts of user defined data types
- f. Support for modular programming

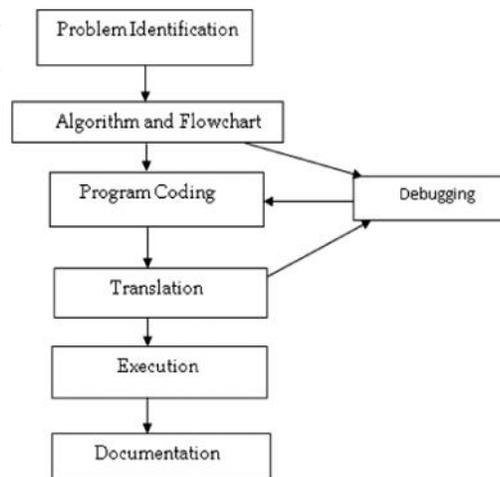
Eg : - Pascal and C

3. Object oriented programming:-Object oriented programming emphasizes on the data rather than the algorithm. Important features of Object oriented programming
 - a. emphasizes on data rather than algorithm
 - b. data abstraction is introduced in addition to procedural abstraction
 - c. Data and associated operations are grouped into single unit
 - d. Programs are designed around the data being operated
 - e. Relationships can be created between similar, yet distinct data types

Eg:- C++, Java

3.1.5 Phases in Programming

The program required can be developed only by going through different stages.



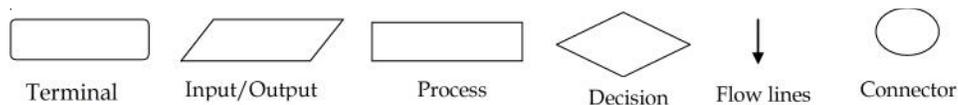
Algorithm

A precise step-by-step procedure to solve a problem is called algorithm.

Flowchart

The pictorial representation of an algorithm is called flowchart.

Flowchart Symbols



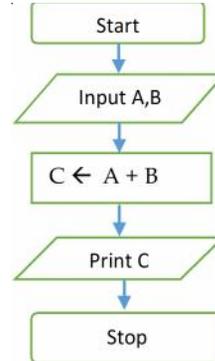
Practical Questions

Give lab work to write algorithm and draw flowchart for the following problems

1. Perform simple mathematical operations

Eg. Algorithm for adding two numbers

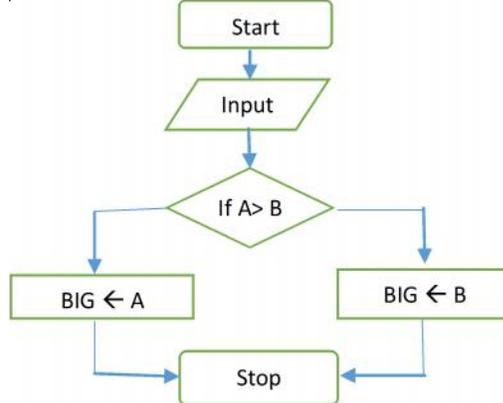
- Step1. Start
 Step2. Input A,B
 Step3. $C \leftarrow A + B$
 Step4. Print C
 Step5. Stop



2. Problems with branching statement

Algorithm for finding the biggest of two numbers

- Step1. Start
 Step2. Input A,B
 Step3. If $A > B$ then
 Step4. $BIG \leftarrow A$
 Else
 Step5. $BIG \leftarrow B$
 Step6. Print BIG
 Step7. Stop

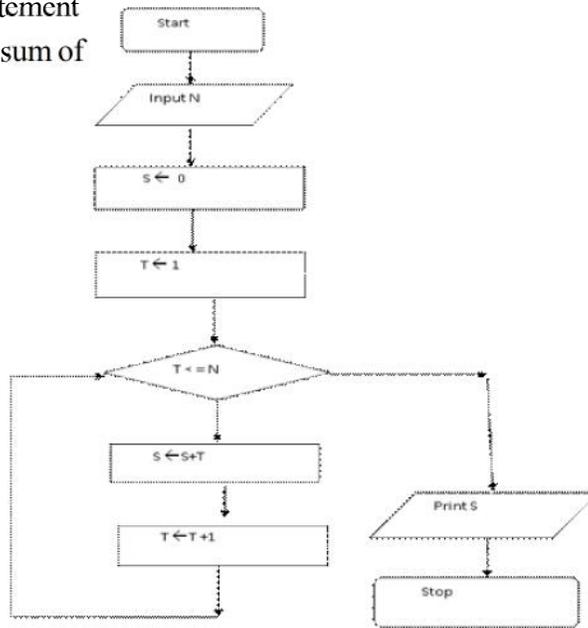


3. Problems with iterative statement

Problems with iterative statement

Algorithm for finding the sum of
 First N natural Numbers

- Step1. Start
 Step2. Input N
 Step3. $S \leftarrow 0$
 Step4. $T \leftarrow 1$
 Step5. Repeat Step 6
 through Step7
 while $T \leq N$
 Step6. $S \leftarrow S + T$
 Step7. $T \leftarrow T + 1$
 Step8. Print S
 Step9. Stop



More Practical Questions

1. Develop an algorithm and draw the flowchart to find sum and average of three numbers
2. Develop an algorithm and draw the flowchart to find simple interest
3. Develop an algorithm and draw the flowchart to find the sum of first N even numbers
4. Develop an algorithm and draw the flowchart to check whether a given number is odd or even
5. Develop an algorithm and draw the flowchart to find the largest of three numbers
6. Develop an algorithm and draw the flowchart to find the factorial of a given number
7. Develop an algorithm and draw the flowchart to find the sum of first N odd numbers
8. Develop an algorithm and draw the flowchart to print the multiplication table of a given number
9. Develop an algorithm and draw the flowchart to input a day number and display the name of the day
10. Based on the evaluation system for standard X, develop an algorithm and draw the flowchart to accept a score out of 100 and find the grade
11. Develop an algorithm and draw the flowchart to input a given number and check whether it is prime or not

TE Questions

Objective Type

1. is a sequence of instructions written in computer language
2. language can be directly interpreted by the hardware
3. "Any complex problem can be solved by breaking it down into different tasks and solving each task by performing simpler activities."
This concept is known as
4. The solution for the main module will be developed only after designing solutions to its sub module. This style of approach is known as

5. A precise step-by-step procedure to solve a problem is called
6. A pictorial representation of an algorithm is called

State True or False

7. 'Computer has no intelligence of its own'
8. 'In Top down design, we start "at the top" with a general problem and design specific solutions to its sub problems.'

Short Answer Type

1. List two popular designing styles of problem solving
2. List the different phases in programming
3. Give the different symbols used in flowchart

Essay Type

1. Compare Top down design and Bottom up design
2. Develop an algorithm and draw the flowchart to input a time in second and convert it into Hr:Min:Sec format (for example if 3700 is given as input , the output should be 1 Hr : 1 Min: 40 sec)
3. Develop an algorithm and draw the flowchart to find the taller one among two students.
4. Compare compiler and interpreter.

Unit - 2

PROGRAMMING IN C++

3.2.1 INTRODUCTION TO C++

C++ was developed by Bjarne Stroustrup starting in 1979 at Bell Labs in Murray Hill, New Jersey. C++ introduces object-oriented programming (OOP) features to C language. It offers classes, which provide the four features commonly present in OOP languages: abstraction, encapsulation, inheritance, and polymorphism. C with Classes later it was renamed C++ in 1983. C++ fully supports object-oriented programming, including the four pillars (abstraction, encapsulation, inheritance, and polymorphism) of object-oriented development.

Learning Outcomes of the Unit

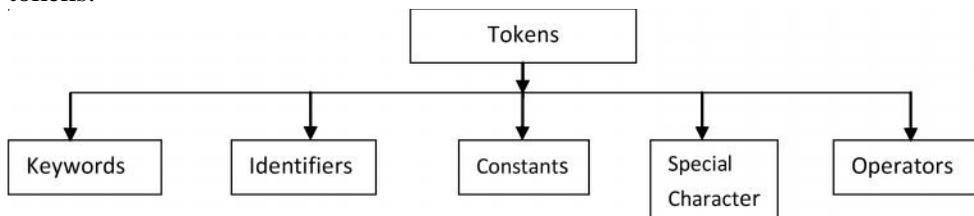
The learner :

- Lists the character set and tokens in C++
- Classifies different data types
- Explains different control statements

3.2.1.1 Character Set & Tokens

Character Set consists of upper and lower case alphabets, digits, special character and white spaces.

C++ programs consist of many elements, which are identified by the compiler as tokens.



Identifier

C++ identifiers a name used to identify a variable, function, class, module, or any other user-defined item. The following rules must remember while naming an identifier

- An identifier starts with a letter A to Z or a to z or an underscore (`_`) followed by zero or more letters, underscores, and digits (0 to 9).
- C++ does not allow punctuation characters such as `@`, `$`, and `%` within identifiers.

- C++ is a case-sensitive programming language. Thus, Manpower and manpower are two different identifiers in C++.
- Here are some examples of acceptable identifiers: mohd, zara, abc, move_name, a_123

Keywords

Keywords are declared by the C++ language and have a predefined meaning. All keywords must be lower case letters.

Eg:- char, int, float, double, void, if, else, while, for , etc...

Variables

A variable is an entity whose value can be changed during program execution.

Variable names are identifiers used to name variables.

Eg :- marks, total_salary ,...etc

Constants or literals

Constants refer to fixed values that the program may not alter and they are also called literals. Constants can be of any of the basic data types and can be divided into Integer Numerals, Floating-Point Numerals, Characters, Strings and Boolean Values.

Character literals

Character literals are enclosed in single quotes. A character literal can be a plain character (e.g., 'x'), an escape sequence (e.g., '\t'). Some examples are given below

Escape sequence	Meaning
\n	new line
\r	carriage return
\t	horizontal tab

Defining constants

There are two simple ways in C++ to define constants:

- The #define Preprocessor
#define identifier value
- The const Keyword
const type variable = value;

Operators

An operator is a symbol that tells the compiler to perform specific mathematical or logical manipulations.

- Arithmetic Operators
- Relational Operators
- Logical Operators

- Bitwise Operators
 - Assignment Operators
 - Conditional Operator
- 1. Arithmetic operators**
 - + - Addition
 - - Subtraction
 - * - Multiplication
 - / - Division
 - % - Modulus
 - 2. Relational operators**
 - < - Less Than
 - > - Greater Than
 - <= - Less than or equal to
 - >= - Greater than or equal to
 - == - Equal to
 - != - Not equal to
 - 3. Assignment Operator**
 - = - Simple assignment
 - += - compound assignment operator
 - 4. Increment and decrement operators**
 - ++ - Increment operator
 - - Decrement operator
 - 5. Logical operators**
 - && - And
 - || - Or
 - ! - Not
 - 6. Conditional Operator (? :)**

Expression1 ? Expression2 : Expression3

Expression1 is evaluated. If it is true, then Expression2 is evaluated. If Expression1 is false, then Expression3 is evaluated .

Eg:- big= a>b ? a : b;

3.2.1.2 DATA TYPES

Primitive data types

<u>Data type</u>	<u>size in bytes</u>	<u>Range</u>
char	1 byte	-127 to 127 or 0 to 255
int	2bytes	-32768 to 32767

longint	4 bytes	-2147483648 to 2147483647
float	4 bytes	
double	8 bytes	

Several of the basic types can be modified using one or more of these type modifiers:

signed, unsigned, short, long

Derived data types

Derived data types are constructed from fundamental data types through some grouping or alteration in the size

Eg:- array, pointer, function

User defined data types

C++ allows programmers to define their own data types

Eg:-struct, enum, union, class

Variable Definition

A variable definition tells the compiler where and how much storage to create for the variable. A variable definition specifies a data type, and contains a list of one or more variables of that type as follows:

Type variable_list;

Eg: int a, b;

Variables can be initialized (assigned an initial value) in their declaration.

Type variable_name = value;

Eg: int a=10, b=25;

Program 2.1

Write a program that asks the user to enter the weight in grams, and then display the equivalent in Kilograms

```
#include <iostream>
using namespace std;
int main()
{
    int weight, kilo, grams;
    cout<<"Enter weight : ";
    cin>>weight;
    kilo = weight/1000;
    grams = weight % 1000;
    cout<<" Kilo = "<<kilo;
    cout<<"Grams = "<<grams;
    return 0;}

```

Integrated Development Environment (IDE)

- Turbo C++ IDE
- Geany IDE

Input/ Output statement

Input statement

```
cin>>variable name;
```

output statement

```
cout<<"message"<< variable name ;
```

3.2.1.3 CONTROL STATEMENTS

Control statements are used to change the normal sequence of execution of a program. Control statements are generally classified into branching statements and looping statements

Branching statements:

1. if statement

An if statement consists of a Boolean expression followed by one or more statements.

Syntax

```
if(boolean_expression)
{
//statement(s)
}
```

Program 2.2

Write a program to find the larger of two numbers

```
#include <iostream>
using namespace std;
int main()
{ int n1,n2,big;
cout<<" Enter two numbers : ";
cin>>n1>>n2;
if(n1>n2)
big=n1;
else
big=n2;
cout<<" Big is "<<big;
return 0;
}
```

If the Boolean expression evaluates to true, then the block of code inside the if statement will be executed. If Boolean expression evaluates to false, then set of code after the end of if statement will be executed.

2. if...else statement

An if statement can be followed by an optional else statement, which executes when the Boolean expression is false.

Syntax

```
if(boolean_expression)
{
```

```

// statement(s)
}
else
{
// statement(s)
}

```

If the Boolean expression evaluates to true, then if block of code will be executed, otherwise else block of code will be executed.

3. if.....else if....else statement

An if statement can be followed by an optional else if...else statement, which is very useful to test various conditions using single if...else if statement.

4. Switch statement

A switch statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each case.

Syntax

The syntax for a switch statement in C++ is as follows

```

switch(expression)
{
case constant-expression1 :
statement(s);
break; //optional case constant-
expression :
case constant-expression2:
statement(s);
break; //optional
// you can have any number of
case statements.
default : //Optional
statement(s);
}

```

Program 2.3

Write a program to check a given character is a vowel or not

```

#include<iostream>
using namespace std;
int main()
{
    char ch;
    cout<< " enter the given character : ";
    cin>>ch;
    switch(ch)
    {
    case 'a' : case 'A'
    case 'e' : case 'E'
    case 'i' : case 'I'
    case 'o' : case 'O'
    case 'u' : case 'U'
    cout<< " The given character is a vowel";
    break;
    default :
    cout<< " The given character is not a vowel";
    }
    return 0; }

```

Looping or Iterative Statements

1. while loop

A while loop statement repeatedly executes a target statement as long as a given condition is true.

Syntax

```
while(condition)
{
statement(s);
}
```

2. do...while loop

Unlike while loops, which test the loop condition at the top of the loop (entry controlled), the

do...while loop checks its condition at the bottom of the loop (exit controlled).

A do...while loop is similar to a while loop, except that a do...while loop is guaranteed to execute at least one time.

Syntax

```
do
{
statement(s);
}while( condition );
```

3. for loop

A for loop is a repetition control structure that allows you to efficiently write a loop that needs to execute a specific number of time.

Syntax

```
for ( init; condition; increment )
{
statement(s);          }
```

Program 2.4

Write a program to sum the digits of a given number

```
#include<iostream>
using namespace std;
int main()
{
int n,s=0,r;
cout<< " enter the given number : ";
cin>>n;
while(n!=0)
{
r = n % 10;
s = s + r;
n = n/10;
}
cout<< " Sum of digits = "<<s;
return 0;}
```

Break statement

When the break statement is encountered inside a loop, the loop is immediately terminated and program control resumes at the next statement following the loop.

Continue statement

The continue statement works somewhat like the break statement. Instead of forcing termination, however, continue forces the next iteration of the loop to take place, skipping any code in between.

Program 2.5

```
Write a program to find the factorial of a given
number
#include <iostream>
using namespace std;
int main()
{ int n,i;
  long int f=1;
  cout<<" Enter the given number : ";
  cin>>n;
  for(i=1;i<=n;i++)
  f=f*i;
  cout<< " Factorial = "<<f;
  return 0; }
```

Assessment

- *Chart containing different keywords*
- *Chart showing different control statements*
- *Assignment on syntax and working of looping statements*
- *Written test on syntax and working of control statements*
- *Practical test - programs using control statements*

Practical Questions

1. Write a program to check whether a given number is odd or even
2. Write a program to find the sum of first N natural numbers
3. Write a program to find the biggest of three numbers.
4. Write a program to check whether a given number is prime or not.
5. Write a program to print the given number in reverse order.
6. Write a program to input the day number and print the day of the week.
7. Write a program to check whether a given year is a leap year or not.
8. Write a program to convert degree Celsius to Fahrenheit.
9. Write a program to print the first N Fibonacci series.

TE Questions**Objective type**

1. Find the odd one out
a) Mouse b) OMR c) C++ d) Joystick
2. State True or False
'C++ is a super set of C language '
3. Pick out OOP language from the following
a) Pascal b) COBOL c) C++ d) C
4. Manu argues that mark and Mark are different identifiers in C++. Do you agree?
5. State True or False :-
'All keywords comprise lower case letters'
6. Pick out unary operator from the following
a) % b) * c) ++ d) /
7. Find the value given by the following expression if n= 11
 $n \% 3$
8. What will be the value of $P = P++ + ++ Q$ where Q is 22 and P=3 initially
9. Name the header file to be included to use cin and cout in programs.
10. What is type casting?
11. Pick out the user defined data types from the following;
12. char b) void c) struct d) double
13. Predict the output of the following code segment if a=10 and b=10 initially
a++;
small = a<b ? a : b;
cout<< small;
14. Predict the output of the following code segment if x = 15 initially
cout<< x++;
15. Predict the output of the following code segment
m=2;
do
{cout<<m;
m+=2;
} while(m <= 20);

16. How many times the following 'for' loop will execute? `for(; ;)`;
17. Select the entry controlled loop from the following
 - a) while
 - b) do...while
 - c) for
 - d) none of these
18. All pre-processor directive statements should begin with
19. The two types of error that may occur in a program is syntax error or semantic error. Name the type of error in the following code segment.
`n = 10;`
`n= n **2;`
20. State True or False
'All C++ program execution starts from main() function'

Short Answer

1. What is a variable? List the two values associated with it?
2. What is the role of the keyword `const`?
3. Explain the two methods of type conversion
4. Classify the following words into identifiers and keywords
`Double`, `mark`, `void`, `total_salary`, `else`, `default`, `WHILE`, `if`
5. What are the rules to form an identifier?
6. Comments are useful to enhance readability and understand ability of a program. Justify this statement with examples
7. Compare `while` and `do...while` statements.
8. Predict the output of the following code segment
`for(i = 1 ; i<=20; i +=2)`
`cout<<i;`
9. Correct the errors in the following code segment
`int main()`
`{ int a;b;`
`cin>>a>>b;`
`}`

Essay Type

1. Write the syntax and working of loops
2. Write the syntax and working of Decision making statements

3.2.2 ARRAYS AND STRINGS

C++ provides a data structure, the array, which stores a fixed-size sequential collection of elements of the same type. An array is used to store a collection of data, but it is often more useful to think of an array as a collection of variables of the same type. Instead of declaring individual variables, such as number0, number1, ..., and number99, you declare one array variable such as numbers and use numbers[0], numbers[1], and ..., numbers[99] to represent individual variables. All arrays consist of contiguous memory locations. The lowest address corresponds to the first element and the highest address to the last element.

Learning Outcomes

The learner:

- Recognises different types of arrays
- Manipulates strings
- Recognises different string manipulation functions

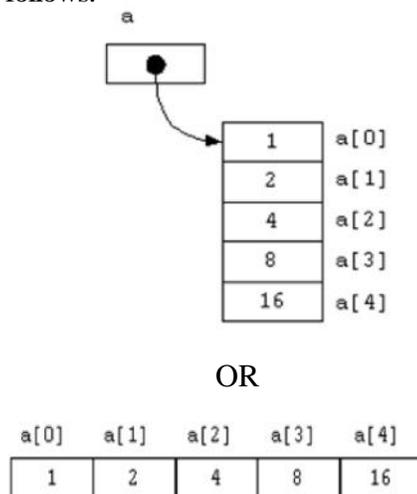
3.2.2.1 Types Of Arrays

Arrays are generally classified into single dimensional array and multi-dimensional array

1. Single Dimensional Array

A single dimensional array consists of a finite linear collection of elements

Declaring Arrays: To declare an array in C++, the programmer specifies the type of the elements and the number of elements required by an array as follows:



Program 2.6

Write a program to read and display the elements of an array with n elements

```
#include <iostream>
using namespace std;
int main()
{
    int n, ar[100],i;
    cout<< " How many elements (< 100 ) ";
    cin>>n;
    for(i=0; i<=n ; i++)
    cin>>ar[i];
    for(i=0; i<=n ; i++)
    cout<<ar[i];
    return 0;
}
```

Type array Name [arraySize];

The arraySize must be an integer constant greater than zero and type can be any valid C++ data type. For example, to declare a 10-element array called balance of type double, use this statement:

```
double balance[10];
```

Initializing Arrays: You can initialize C++ array elements either one by one or using a single statement as follows:

```
double balance[5] = { 1000.0, 2.0, 3.4, 17.0, 50.0};
```

If you omit the size of the array, an array just big enough to hold the initialization is created. Therefore, if you write:

```
double balance[] = { 1000.0, 2.0, 3.4, 17.0, 50.0};
```

Accessing Array Elements: An element is accessed by indexing the array name. This is done by placing the index of the element within square brackets after the name of the array. For example:

```
double salary = balance[9];
```

The above statement will take 10th element from the array and assign the value to salary variable.

2. Multi-dimensional Arrays

The Multi Dimensional Array is an array with two or more index values. It is also known as array of arrays.

The general form of a multidimensional array declaration:

```
type name[size1][size2]...[sizeN];
```

The simplest form of the multidimensional array is the two-dimensional array.

Eg: `int arr[10][10];`

For example, the following declaration creates a three dimensional integer array:

```
Int three dim[5][10][4];
```

3.2.2.2 STRING

The string is an one-dimensional array of characters which is terminated by a null character '\0'. Thus a null-terminated string contains the characters that comprise the string followed by a null.

Program 2.7

Write a program to read and display the elements of a table consists 2 rows and 3 columns

```
#include <iostream>
using namespace std;
int main()
{
    Int tb[2][3],i,j;
    cout<< " Enter the elements ";
    for(i=0; i<2 ; i++)
    for(j=0; j<3 ; j++)
    cin>>tb[i][j];
    cout<< " The elements are :";
    for(i=0; i<2 ; i++)
    for(j=0; j<3 ; j++)
    cout<<tb[i][j];
    return 0;
}
```

The following declaration and initialization create a string consisting of the word "Hello".

```
char greeting[6] = {'H', 'e', 'l', 'l', 'o', '\0'};
```

If you follow the array initialization, then you can write the above statement as follows:

```
char greeting[] = "Hello";
```

C++ supports a wide range of functions that manipulate null-terminated strings

Eg:- `strcpy()`, `strcat()`, `strlen()`, `strcmp()`,

Assessment

- *Chart showing different types of arrays*
- *Lab Work on matrix manipulation (linear search and bubble sort)*
- *Assignment on different string functions*
- *Written test that contains different aspects of single dimensional array*

Practical Questions

1. Write a program to read and display the elements of the array `intar[50]`;
2. Write a program to display the elements of the following array in reverse order
in `tarr[] = { 23,12,3,6,9,67,45,5,8,14,35,56,87 }`;
3. Write a program to read and display the elements of a table with `n` rows and `m` columns
4. Write a program to count the number of characters in the following string
" Computer Science and Information Technology "
5. Write a program to add 5 to each element of an array with `N` elements.
6. Write a program to find the biggest element in an array with `n` elements.

TE Questions

Objective type

1. Which of the following is the correct declaration of an array?
a) `int m (10)` b) `int [10] m` c.) `m [10] int` d) `int m [10]`
2. Identify the last index of the array `float P[8]`; from the following choices.
a. 7 b. 8 c. 0 d. 9
3. Consider the following C++ statements:
`int A[4] = {1,2,3,4};`
`int B = (A[3]+A[0]+A[2]) / A[1];`
Predict the value of the variable `B`.

5. Consider the following array declaration,
`int A[]={4,5,8} ; int B[]={2,10};`
Write a valid C++ statement for finding the difference between the last element of the array B and the first element of the array A.
6. Write a statement for storing the string "NO SMOKING" using a character array.
7. Suppose NUM is an array containing 10 integer numbers. Identify the name of the operation in which all the elements of the array are increased by 1 with the help of ++ operator.
8. If `int M[20];` is an array contains 1 to 20 natural numbers, then which element of the array will be referenced by `M[11]` ?
9. How many bytes will be allocated in the memory to store the string "MY SCHOOL"?
10. What will be the output of the following code segment if the input is "ComputerScience"?
`char WORD[15];
cin>>WORD;
cout<<WORD;`
11. The following code segment does not give 6 as the output. Why?
`char N[]="123";
cout<<(N[0]+N[1]+N[2]);`

Short answer type

1. Define an array and write the syntax of declaring a single dimensional array
2. Write the string function used for string copy and concatenate
3. Correct the error in the following code segment
`int n =50;
intarr[n];`

Essay Type

1. Write a program to read and display elements of a single dimensional array with n elements
2. Write a program that counts number of vowels and consonants in a line of text
3. Write a program to read and display elements of a table with 5 rows and 4 columns.

3.2.3 STRUCTURE IN C++

C++ arrays allow you to define variables that combine several data items of the same kind, but structure is another user defined data type which allows you to combine data items of different kinds.

Structures are used to represent a record, suppose you want to keep track of your books in a library. You might want to track the following attributes about each book:

Title, Author, Subject, Book ID

Learning Outcomes

The learner:

- Explains structure
- Identifies nested structure
- Develops programs using array of structure

3.2.3.1 Defining a Structure

Structure is a heterogeneous (different types) collection of elements. To define a structure, you must use the struct statement. The struct statement defines a new data type, with more than one member, for your program. The format of the struct statement is this:

```
struct [structure tag]
{
member definition;
member definition;
...
member definition;
} [one or more structure variables];
```

The structure tag is optional

Program 2.8

Write a program to read the details of an employe

```
#include <iostream>
using namespace std;
struct employe
{
int empno;
char name[30];
char job[25];
long int salary;
int age;
};
int main()
{
employe e;
cout << "Enter the details of an Employee";
cin >> e.empno;
cin >> e.name;
cin >> e.job;
cin >> e.salary;
cin >> e.age;
return 0;
}
```

Accessing Structure Members

To access any member of a structure, we use the member access operator (.)

Eg:-

```
struct time
{
    int hr;
    int min;
    int sec;
};

time t;
cout<<t.hr<<t.min<<t.sec
```

3.2.3.2 Nested Structure

A structure may consist of structures inside it, which is known as nested structure.

Eg :-

```
struct date
{
    int day;
    int month;
    int year;
};

struct student
{
    int roll_no;
    char name[30];
    struct date dob;
    char course[25];
    int marks;
};
```

3.2.3.3 Array of Structure

Array of structure is a finite collection of structure variables

Syntax:

Structure name array name[size];

Eg: employee [50];

Here 'e' is an array contains 50 elements of type employe structure

Program2.9

Write a program to read and display the details of N students in a class(array of structure)

```
#include <iostream>
using namespace std;
struct student
{
    int adno;
    char name[30];
    char course[25];
    int age;
};

int main()
{
    student s[60];
    int n, i;
    cout<< " How many students ( < 60 ) ";
    cin>>n;
    for(i=0; i<=n ; i++)
        {   cin>> s[i].adno;
            cin>>s[i].name; cin>> s[i].course;
        }
    for(i=0; i<=n ; i++)
        {   cin>> s[i].adno;
            cin>>s[i].name;
            cin>> s[i].course;
        }
    return 0;
}
```

Assessment

1. *Lab Work on structure manipulations*
2. *Activity log preparation on structure definition and accessing elements of structure*
3. *Written test on different aspects on structure*

Practical Questions

1. Write a program to read and display the details of a student (use appropriate members)
2. Write a program that illustrates nested structure.
3. Write a program to read and display the details of 100 Employees in a company.

TE Questions**Objective Type**

1. The members of a structure are accessed using Operator
2. What is structure tag?
3. A structure inside another structure is called
4. Predict the output of the following code segments

```
struct ABC
{
    int a;
    float b;
    double c;
} ob;
cout<<sizeof(ob);
```

Short Answer Type

1. Explain storage organization of structure variable
2. Why structures are called heterogeneous data type?
3. Write the syntax of nested structure
4. Write the syntax of defining an array of structures

Essay Type

1. Write a program to read and display the details of N students. A student record has elements such as rollno, name, date of birth (nested structure), sex, course

3.2.4 FUNCTIONS

A function is a group of statements that together perform a task. Every C++ program has at least one function, which is `main()`, and all the most trivial programs can define additional functions. A function is generally classified into built-in function and user defined function.

Learning Outcomes

The learner:

- Lists different built-in functions
- Recognises different aspects of user defined functions
- Demonstrates the different parameter passing methods
- Explains the concept of pointers and operators '&' and '*'

3.2.4.1 Built-in Functions

A built in function is a predefined function which is included in various C++ header files.

Eg.: `clrscr()`, `getch()`, `sqrt()`, `pow()`, `strcpy()`, `setw()`etc

Built-in function	Use	Header files
<code>clrscr()</code>	To clear the output screen	<code>conio.h</code>
<code>getch()</code>	To read a character from the keyboard	<code>conio.h</code>
<code>sqrt()</code>	To get the square root of a number	<code>math.h</code>
<code>pow()</code>	To get the power of a given number	<code>math.h</code>
<code>strcpy()</code>	To copy a string to another string	<code>string.h</code>
<code>setw()</code>	To set the space between output values	<code>process.h</code>

3.2.4.2 User Defined Function

Defining a Function

A function definition provides the actual body of the function. C++ function definition consists of a function header and a function body.

Syntax

```
return_type function_name( parameter list )
{
    body of the function
}
```

Return Type: A function may return a value. The return type is the data type of the value the function returns. Some functions perform the desired operations without returning a value. In this case, the return_type is the keyword void. Default return type is 'int'

Function Name: This is the actual name of the function. The number and type of the parameter list together constitute the function signature.

Parameters: The values mentioned in the bracket of a function are called parameters or arguments. The parameter list refers to the type, order, and number of the parameters of a function. Parameters are optional; that is, a function may contain no parameters. There are two types of Parameters or Arguments

- a. Formal parameters: Parameters used in function definition.
- b. Actual Parameters: Parameters used in function call

Function Body: The function body contains a collection of statements that define what the function does.

Function Declaration/ Prototype: A function declaration tells the compiler about a function name and

how to call the function. The actual body of the function can be defined separately. A function declaration has the following parts:

```
return_type function_name (parameterlist );
```

Function declaration is not necessary if you write function body before function call.

Program 2.10

Write a program to swap the content of two memory locations (call-by-value)

```
#include <iostream>
using namespace std;
void swap( int x, int y );
int main()
{
    int a, b;
    cout<< " Enter two numbers ";
    cin>>a>>b;
    swap(a,b);
    cout<< " The values after exchange ";
    cout<<a<<b;
    return 0;
}
void swap( int x, int y )
{
    int t;
    t=x;
    x=y;
    y=t;
}
```

3.2.4.3 Parameter passing:

Call by Value: The call by value method of passing arguments to a function copies the actual value of an argument into the formal parameter of the function.

By default, C++ uses call by value to pass arguments. In general, this means that code within a function cannot alter the arguments used to call the function.

Call by Reference: The call by reference method of passing arguments to a function copies the reference of an argument into the formal parameter. Inside the function, the reference is used to access the actual argument used in the call. This means that changes

made to the formal arguments affect the actual arguments.

To pass the value by reference, argument reference is passed to the functions just like any other value.

3.2.4.4 Pointers

A pointer is a variable which holds the address of another variable. Like any variable or constant, you must declare a pointer before you can work with it. The general form of a pointer variable declaration is:

```
type * var_name;
```

Here, type is the pointer's data type; it must be a valid C++ type and var_name is the name of the pointer variable. However, in this statement the asterisk is being used to designate a variable as a pointer. Following are the valid pointer declarations.

```
int *ip; // pointer to an integer
```

```
double *dp; // pointer to a double
```

```
float *fp; // pointer to a float
```

```
char *ch // pointer to character
```

Program 2.11

Write a program to swap the content of two memory locations (call-by-reference)

```
#include <iostream>
using namespace std;
void swap( int &x, int &y);
int main()
{
int a, b;
cout<< " Enter two numbers ";
cin>>a>>b;
swap(a,b);
cout<< " The values after exchange ";
cout<<a<<b;
return 0;
}
void swap( int &x, int &y )
{ int t;
t=x;
x=y;
y=t;
}
```

Address operator (&)

The address operator return the base address of a variable. It is a unary operator placed before a variable.

```
Eg: int *p;
      int a;
      p=&a;
```

Use of *(Redirection) operator

There are few important operations with *.

- (a) We define a pointer variable.
- (b) Access the value at the address available in the pointer variable.

This is done by using unary operator * that returns the value of the variable located at the address specified by its operand.

Assessment

- *Chart showing different built-in functions*
- *Seminar on user defined functions*
- *Lab Work on programs using user defined functions*
- *Written test on user defined functions*

Practical Questions

1. Write the function sum() with four parameters that calculates the arguments provided and returns their sum.
2. The factorial n! of a positive integer n is defined as

$$n! = 1 * 2 * 3 \dots * (n-1) * n$$
 Where $0! = 1$
3. Write a function to calculate the factorial of a number.
4. Write a program to increment the value of two variables by 1
 - a) Using call-by-value
 - b) Using call-by-reference

TE Questions**Objective Type**

1. _____ function can be used to find the square root of a number.

2. A function returns _____ values.
a. zero b. one c. zero or one d. none of these
3. The default return type of a function is
4. Pick out the address operator from the following
a. * b. && c. & d. sizeof()
5. is a redirection operator
6. State True or False
"A function returns a maximum of only one value."

Short Answer Type

1. Define actual parameter and formal parameter
2. Write down the prototype of a function that can find the sum of 2 numbers.
3. What does the following function do?

```
void example(int n)
{
    int i;
    for (i=0; i<n; i++)
        cout<< '*';
}
```

4. Observe the following function prototypes:
int fun(int);
int fun(int&);
How do these functions differ in the method of calling?
5. Define actual parameter and formal parameter

Essay Type

1. Write a C++ program to find the factorial of a given number using a function
2. Briefly explain call-by-value and call-by-reference with the help of a program

3.2.5 OBJECT ORIENTED PROGRAMMING

The main purpose of C++ programming is to add object orientation to the C programming language and classes are the central feature of C++ that supports object-oriented programming and are often called user-defined types.

A class is used to specify the form of an object and it combines data representation and methods for manipulating that data into one neat package. The data and functions within a class are called members of the class.

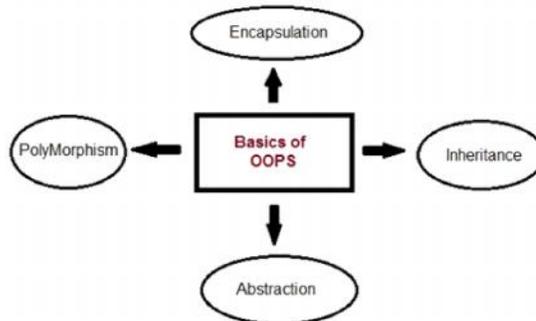
Learning Outcomes

The learner:

- Describes the different Object Oriented Programming concepts
- Explains class and objects
- Differentiates member functions
- Illustrates the use of friend functions and friend class
- Explains constructors and destructors

3.2.5.1 Features of Oop

Object oriented programming is a programming style that is associated with the concept of objects, having datafields and related member functions.



Main Features of Object Oriented Programming

- Class
- Object
- Modularity
- Abstraction
- Encapsulation

- Inheritance
- Overloading
- Exception Handling

Class

A Class is a construct in C++ which is used to bind data and associated function together in to a single unit. Class is a user defined data type.

Eg : We consider human body as a class, we do have multiple objects of this class, with variable as colour, hair etc. and methods as walking, speaking etc.

Objects

Objects are the basic unit of OOP. They are instances of class, which have data members and uses various member functions to perform tasks.

Modularity

Modularity is designing a system that is divided into a set of functional units (named modules) that can be composed into a larger application.

Abstraction

Abstraction refers to showing only the essential features without revealing background details.

Encapsulation

Encapsulation is all about binding the data variables and functions together in class. It can also be called data binding.

Inheritance

Inheritance is the technique of building new classes (derived class) from an existing class (base class). The most important advantage of inheritance is code reusability.

Polymorphism

Polymorphism is the ability for a message or function behaves differently depending upon the situation.

Overloading

Overloading is a part of polymorphism. Where a function or operator is made & defined many times, to perform different functions they are said to be overloaded.

Exception Handling

Exception handling is a feature of OOP, to handle unresolved exceptions or errors produced at runtime.

3.2.5.2 Class And Objects

Class: A class definition starts with the keyword `class` followed by the class name; and the class body, enclosed by a pair of curly braces. A class definition must be followed either by a semicolon or a list of declarations. For example, we define the `Box` data type using the keyword `class` as follows:

```
class Box
{
public:
double length; // Length of a box
double breadth; // Breadth of a box
double height; // Height of a box
};
```

The keyword `public` determines the access attributes of the members of the class that follows it. A public member can be accessed from outside the class anywhere within the scope of the class object. You can also specify the members of a class as private or protected which we will discuss in a sub-section

Objects

A class provides the blueprints for objects, so basically an object is created from a class. We declare objects of a class with exactly the same sort of declaration that we declare variables of basic types.

Syntax:-

```
ClassName ObjectName;
```

Eg:-

```
Box b;
```

Program 2.12

Write a program to read and display the details of an employee

```
#include <iostream>
using namespace std;
class employe
{ private :
Intempno;
char name[30];
char designation[25];
longint salary;
public :
void ReadData()
    { cin >> empno;
cin >> name;
cin >> designation;
cin >> salary;
    }
Void DispData()
    { cout << empno;
cout << name;
cout << designation;
cout << salary;
    }
};
int main()
{ employe e;
cout << "Enter details of an employe ";
e.ReadData();
cout << " The details are : ";
e.DispData();
return 0;
```

Class Access Modifiers

Data hiding is one of the important features of Object Oriented Programming which allows preventing the functions of a program to access directly the internal representation of a class type. The access restriction to the class members is specified by the labeled public, private, and protected sections within the class body. The keywords public, private, and protected are called access specifiers.

A class can have multiple public, protected, or private labeled sections. Each section remains in effect until either another section label or the closing right brace of the class body is seen. The default access for members and classes is private.

```
class classname
{
public:
// public members go here
protected:
// protected members go here
private:
};
```

The Public Members

A public member is accessible from anywhere outside the class but within a program. You can set and get the value of public variables without any member function.

The Private Members

A private member variable or function cannot be accessed, or even viewed from outside the class. Only the class and friend functions can access private members. By default all the members of a class would be private.

The Protected Members

A protected member variable or function is very similar to a private member but it provided one additional benefit that they can be accessed in child classes which are called derived classes.

Accessing the Data Members

The public data members of objects of a class can be accessed using the direct member access operator (.).

Eg:-

```
class box
{
    private :
        int length;
    public :
        int breadth;
};
box b;
b.breadth;
b.length; // error, can not access private members using dot operator.
```

You will learn derived classes and inheritance in next chapter.

3.2.5.3 Class Member Functions

A member function of a class is a function that has its definition or its prototype within the class definition like any other variable. It operates on any object of the class of which it is a member, and has access to all the members of a class for that object.

Inline Member Functions

C++ inline function is powerful concept that is commonly used with classes. If a function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time.

To inline a function, place the keyword inline before the function name and define the function before any calls are made to the function. The compiler can ignore the inline qualifier in case defined function is more than a line.

A function definition in a class definition is an inline function definition, even without the use of the inline specifier.

Outline Member Functions

Member function definition outside the class. Scope resolution operator (::) is used for this purpose.

3.2.5.4 Friend Function and Friend Class

Friend function: A friend function is a non-member function which access the private and protected members of the class in which it is declared as friend.

To declare a function as a friend of a class, precede the function prototype in the class definition with keyword friend as follows:

```
class Box
{
double width;
public:
double length;
friend void printWidth( Box box );
void setWidth( double wid );
};
void printWidth( Box box ) // Note: printWidth() is not a member function of any
class.
{
/* Because printWidth() is a friend of Box, it can
directly access any member of this class */
cout<< "Width of box : " <<box.width<<endl;
}
int main() // Main function for the program
{
Box box; // set box width without member function
box.setWidth(10.0); // Use friend function to print the width.
printWidth( box );
return 0;
}
```

Friend Class: To declare all member functions of class ClassTwo as friends of class ClassOne, place a following declaration in the definition of class ClassOne.

Friend class ClassTwo;

3.2.5.5 Constructor & Destructor

A class constructor is a special member function of a class that is executed whenever we create new objects of that class.

A constructor will have exact same name as the class and it does not have any return type at all, not even void. Constructors can be very useful for setting initial values for certain member variables.

Characteristics of constructors:

1. Its name is same as that of the class
2. It does not have any return type at all, not even void.
3. A constructor can be over-loaded
4. It is used to initialize data members
5. It must be declared in the public part

Different types of constructors

1. Default constructor: It is a constructor with no arguments
2. Parameterized constructor: It is a constructor with arguments
3. Copy constructor: The copy constructor is used to initialize one object from another of the same type. Copy an object to pass it as an argument to a function.

Destructor: A destructor is a special member function of a class that is executed whenever an object of its class goes out of scope or whenever the delete expression is applied to a pointer to the object of that class.

A destructor will have exact same name as the class prefixed with a

Program 2.13

Write a program used to display the applications of constructor

```
#include <iostream>
using namespace std;
class Game
{
private:
int goals;
public:
// constructor used to initialize
Game() {
goals = 0;
}
// return score
int getGoals() {
return goals;
}
// increment goal by one
void incrementGoal() {
goals++;
}
};
int main() {
Game football;
cout<< "Number of goals when game is
started = " <<football.getGoals() <<endl;
football.incrementGoal();
football.incrementGoal();
cout<< "Number of goals a little later = "
<<football.getGoals() <<endl;
return 0;
}
```

tilde (~) and it can neither return a value nor can it take any parameters. Destructor can be very useful for releasing resources before coming out of the program like closing files, releasing memories etc.

Practical Questions

1. Create a class student. Data members : adno, name, sex, age, course
Member functions: ReadData() to read data members and DispData() to display data members
2. Write a program that illustrates the implementation of different types of constructors
3. Write a program that illustrates the implementation of destructor
4. Write a program that implement the following class
ClassName: employee
Datamembers: empnum, name, age, designation, salary
Member functions(outline) : ReadData() to read data members and DispData() to display data members

TE Questions

1. C++ is a _____ paradigm language.
a. Object oriented b. Procedural c. Both a and b d. None of the above.
2. C, Pascal, FORTRAN and BASIC are examples of _____ languages.
a. Object oriented b. Procedural c. Both a and b d. None of the above
3. To reduce the complexity, in procedural paradigm the functions associated with a common task are grouped into _____.
a. Structures b. Programs c. Modules d. Interface.
4. Data is undervalued in _____ paradigm.
a. Object oriented b. Procedural c. Both a and b d. None of the above.
5. _____ is a prototype/blue print that defines the specification common to all objects of a particular type.
a. Object b. Class c. Method d. Interface.
6. _____ of the class are called individual instances.
a. Object b. Class c. Method d. Interface.

7. Based on a class _____ number of objects can be created.
a. 3 b. 2 c. Any d. 1.
8. Calling member function of an object from another object is called_____.
a. Sending letter b. Calling function c. Passing message d. Switching
9. In object-oriented programming, _____ refers to a programming language's ability to process objects differently depending on their data type or class.
a. Encapsulation b. Abstraction c. Inheritance d. Polymorphism
10. The process of giving different tasks to functions with the same name, but different signatures is called _____.
a. Inheritance b. Function overloading c. Passing message
d. Switching
11. State True or False
'A constructor can be overloaded'
12. State True or False
'A friend function is a non member function'

Short Answer Type

1. What is a class? Can one create any number of objects from a class?
2. What is an Object? Give an example.
3. What are the basic concepts of OOP?
4. What is encapsulation?
5. What is the relationship between a class and an object?
6. What is the difference between data encapsulation and abstraction?
7. Define a constructor. Write its characteristics
8. Define a destructor. Write its characteristics
9. Briefly explain the different types of constructors

Essay Type

1. Describe the features of object oriented programming.
2. What are the advantages of OOP?
3. A friend function act as a bridge between two classes. Substantiate with the help of a program
4. Write a program that make use of different types of constructors

3.2.6 POLYMORPHISM & INHERITANCE

Polymorphism

C++ allows you to specify more than one definition for a function name or an operator in the same scope, which is called function overloading and operator overloading respectively.

An overloaded declaration is a declaration that is declared with the same name as a previously declared declaration in the same scope, except that both declarations have different arguments and obviously different definition (implementation).

When you call an overloaded function or operator, the compiler determines the most appropriate definition to use, by comparing the argument types you have used to call the function or operator with the parameter types specified in the definitions. The process of selecting the most appropriate overloaded function or operator is called overload resolution

Learning Outcomes

The learner:

- Explains polymorphism
- Demonstrates the implementation of Polymorphism
- Describe inheritance

3.2.6.1 Polymorphism

Polymorphism means that some code or operations or objects behave differently in different contexts.

For example, the + (plus) operator in C++:

Program 2.14

Following is the example to show the concept of binary operator over loading (+) using a member function.

```
#include <iostream>
using namespace std;
class complex
{ private :
float real; float imag;
public :
complex()
{real = imag = 0.0; }
void getdata()
{ cout<< " Real Part : ";cin>>real;
cout<< "Imaginary Part : ";cin>>imag;}
void dispdata()
{cout<< " Real Part = " <<real;
cout<< "Imaginary Part " <<imag;}
complex addcomplex ( complex c2);};
complex addcomplex ( complex c2)
{complex temp;
temp.real = real + c2.real;
temp.imag = imag + c2.imag;
return(temp);}
int main()
{complex c1,c2,c3;
cout<< " Enter complex number 1
";c1.getdata();
cout<< " Enter complex number 2
";c2.getdata();
c3=c1.addcomplex(c2);c3.dispdata();return
0;
}
```

4 + 5 - integer addition

3.14 + 2.0 - floating point addition

s1 + "bar" - string concatenation!

In C++, there are two types of polymorphism - Compile time polymorphism and run time polymorphism

3.2.6.2 Implementation of Polymorphism

Function Overloading in C++

You can have multiple definitions for the same function name in the same scope. The definition of the function must differ from each other by the types and/or the number of arguments in the argument list. You cannot overload function declarations that differ only by return type.

Operators Overloading in C++

You can redefine or overload most of the built-in operators available in C++. Thus, a programmer can use operators with user-defined types as well. Overloaded operators are functions with special names the keyword operator followed by the symbol for the operator being defined.

Syntax :

Classname operator + (constclass name &, constclassname&);

3.2.6.3 Inheritance

One of the most important concepts in object-oriented programming is that of inheritance. The technique of building new class(s) from an existing class(s) is called inheritance. This also provides an opportunity to reuse the code functionality and fast implementation time. When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the base class, and the new class is referred to as the derived class.

The idea of inheritance implements the IS-A relationship. For example, mammal IS-A animal, dog IS-A mammal hence dog IS-A animal as well and so on.

Type of Inheritance

1. Single Inheritance

Derivation of a class from only one base class is called single inheritance

2. Multiple Inheritance

Derivation of a class from several base class.

3. Hierarchical Inheritance

Derivation of several classes from a single base class

4. Multilevel Inheritance

Derivation of a class from another derived class

5. Hybrid Inheritance

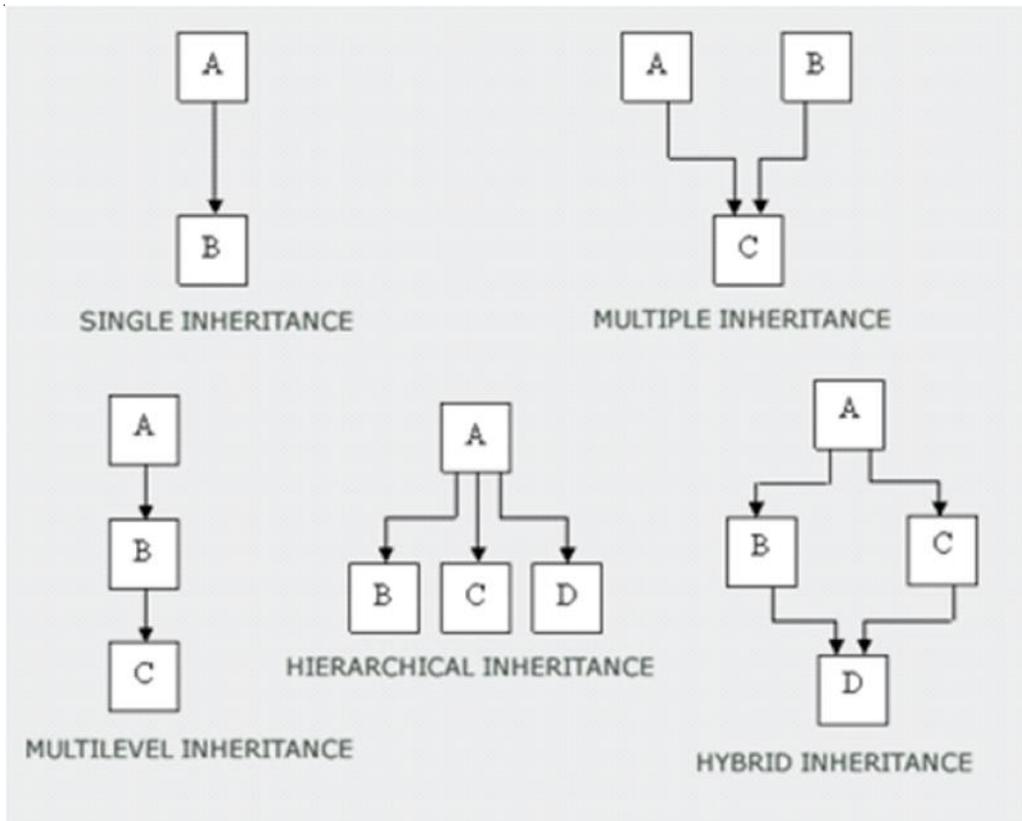
Derivation of a class involving more than one form of inheritances

Program 2.15

Implementation of Single Inheritance

```
#include <iostream>
using namespace std;
class Shape
{
public:
voidsetWidth(int w)
{
width = w;
}
voidsetHeight(int h)
{
height = h;
}
protected:
int width;
int height;
};
```

```
class Rectangle: public
Shape
{
public:
intgetArea()
{
return (width * height);
}
};
int main(void)
{
Rectangle Rect;
Rect.setWidth(5);
Rect.setHeight(7);
// Print the area of the
object.
cout<< "Total area: "
<<Rect.getArea() <<endl;
return 0;
}
```



Single Inheritance

Syntax :

class derived-class: [visibility mode] base-class;

Where visibility mode is one of public, protected, or private, and base-class is the name of a previously defined class. If the visibility mode not used, then it is private by default.

Member Access Specifier	How Members of the Base Class Appear in the Derived Class
Private	Private members of the base class are inaccessible to the derived class. Protected members of the base class become private members of the derived class. Public members of the base class become private members of the derived class.
Protected	Private members of the base class are inaccessible to the derived class. Protected members of the base class become protected members of the derived class. Public members of the base class become protected members of the derived class.
Public	Private members of the base class are inaccessible to the derived class. Protected members of the base class become protected members of the derived class. Public members of the base class become public members of the derived class.

Access Control and Inheritance

A derived class can access all the non-private members of its base class.

We can summarize the different visibility modes according to - who can access them, in the following way:

Access	public	protected	private
Same class	yes	yes	yes
Derived classes	yes	yes	no
Outside classes	yes	no	no

Multiple Inheritances

A C++ class can inherit members from more than one class

Syntax:

class derived-class: [visibility mode] baseA, [visibility mode] baseB....

Practical Questions

Assessment

- *Lab Work on programs using function overloading and operator overloading*
- *Lab Work on programs using single inheritance and multiple inheritance*
- *Chart contains different types of inheritance*
- *Seminar on function overloading and operator overloading*
- *Written test on polymorphism and inheritance*

Practical Questions

1. Design a program that illustrate single inheritance
2. Design a program that illustrate multiple inheritance

TE Questions

Objective Type

1. number of definitions can be attached to a function name
a) 0 b) 1 c) 2 d) None of these
2. Deriving a subclass from multiple base class is called inheritance
a) single b) multiple c) hierarchical d) multilevel
3. While using Visibility mode , the public members of base class become public to the derived class
a) public b) private c) protected d) either a or b
4. The function `int add(int,float);` matches with
a) `add(20)` b) `add(10,15.5)` c) `add(13.5)` d) none of these

Short answer Type

1. What is function overloading?
2. What is meant by signature?
3. What is meant by constructor overloading?
4. Define polymorphism.
5. Compare derivation and friendship
6. What is operator overloading?

Essay Type

1. Write a program to illustrate the working of overloaded functions.
2. What is Inheritance? Explain different types of inheritance
3. Write a program to illustrate the working of single inheritance

3.2.7 FILES IN C++

Files are used to store information or data permanently in computer memory. File is a collection of related records. Files are generally classified into sequential access file and Random access file.

Learning Outcomes

- Explains different file stream classes
- Lists different file modes for opening a file
- Describe different file operations
- Lists the different error handling functions

3.3.7.1 FILE STREAM CLASSES

So far, we have been using the iostream standard library, which provides cin and cout methods for reading from standard input and writing to standard output respectively. Here we describe different file stream classes

File stream	Description
ofstream:	This file stream represents the output file stream and is used to create files and to write information to files.
ifstream:	This file stream represents the input file stream and is used to read information from files.
fstream:	This file stream represents the file stream generally, and has the capabilities of both ofstream and ifstream which means it can create files, write information to files, and read information from files.

To perform file processing in C++, header files <iostream> and <fstream> must be included in your C++ source file.

3.2.7.2 Opening A File

A file must be opened before you can read from it or write to it. Either of stream or fstream object may be used to open a file for writing. And ifstream object is used to open a file for reading purpose only. Following is the standard syntax for open() function, which is a member of fstream, ifstream, and ofstream objects.

Streamobject.open (" filename", filemode);

File Modes	Description
ios::app	Append mode. All output to that file to be appended to the end.
ios::ate	Open a file for output and move the R/W control to the end of the file.
ios::in	Open a file for reading.
ios::out	Open a file for writing.
ios::trunc	If the file already exists, contents will be truncated

Closing a File

When a C++ program terminates it automatically flushes all the streams, release all the allocated memory and close all the opened files. But it is always a good practice that a programmer should close all the opened files before program termination.

Syntax

```
Streamobject.close();
```

3.2.7.3 File Operations

Writing to a File: While doing C++ programming, you write information to a file from your program using the stream insertion operator (<<) just as you use that operator to output information to the screen. The only difference is that you use an ofstream or fstream object instead of the cout object.

The write() function also used to write information to a file

Syntax

```
File stream object.write((char*)& variable,
size of (variable));
```

Reading from a File: You read information from a file into your program using the stream extraction operator (>>) just as you use that operator to input information from the keyboard. The only difference is that you use an ifstream or fstream object instead of the cin object.

The read() function also used to read information from a file

Syntax

```
Filestreamobject.read( (char *)& variable,
sizeof(variable));
```

Error handling functions

- eof(): True if end of file is encountered, otherwise false
- fail(): True if read or write operation has failed, otherwise false
- clear(): Clear error states and further operations can be attempted
- good(): True if operation is successful
- bad(): True if invalid operation attempted

Program 2.16

Write a program to write the details of student to a file "vhse.dat"

```
#include <iostream>
#include <fstream>
using namespace std;
struct student
{ int adno;
char name[30];
char course[25];
int age;
};
int main()
{ struct s;
ofstream outfile;
outfile.open("vhse.dat", ios::out)
cout<< " Enter details of a student";
cin>>s.adno;
cin>>s.name;
cin>>s.course;
outfile.write( (char *) & s, sizeof(s));
outfile.close();
```

Assessment

- Chart showing different file modes
- Lab Work on program using write() function
- Lab Work on program using read() function

Practical Questions

1. Write a program to write a line of text to a file "line.txt"
2. Write a program to write the details of n students to a file "vhse.dat" using structure
3. Write a program to write the details of n students to a file "csit.dat" using class (use appropriate data members and member functions)
4. Write a program to read the details in the file "vhse.dat" using structure
5. Write a program to read the details in the file "csit.dat" using class(use appropriate data members and member functions)

TE Questions**Objective Type:**

1. State True or False
'A filename should have primary name '
2. is a collection of related records
3. State True or False
'A file can be opened by using a constructor'

Short Answer Type

1. Write three important file stream classes needed for file manipulations
2. Describe different methods of opening a file
3. Match the following

ios :: in	Truncate the file if it already exists
ios :: out	All writes occur at end of file
ios :: app	Open for reading
ios :: trunc	Open for writing

4. Match the following

eof()	TRUE if operation is successful else FALSE
fail()	TRUE if invalid operation attempted else FALSE
bad()	TRUE if read or write operation has failed else FALSE
good()	TRUE if end of file is reached otherwise FLASE

Essay Type

1. What is a file? What are the steps involved in manipulating a file in C++ program?
2. Describe different methods of opening a file. Write a program to open a file named "my.bio" and write your name and other details.

Unit - 3

DATABASE DESIGNING

Database Management System or DBMS refers to the technology of storing and retrieving data with maximum efficiency along with appropriate security measures. This unit explains the basics of DBMS such as its architecture, data models, data schemas, data independence, relational database design, and storage and file structure. The limitations of conventional file system and advantages of DBMS are also discussed. This unit also explores the concept of data abstraction, data independence and main terminologies in Relational Database Management System (RDBMS).

Learning Outcomes

The learner

- Understands the concept of DBMS
- Identifies advantages of the database management system over conventional file management system.
- Explains data abstraction and data independence in DBMS.
- Lists and explains the data models in DBMS.
- Identifies different terminologies in RDBMS.
- Recognises Keys in RDBMS

3.3.1 Database Management System (DBMS)

Data: Raw facts and figures are called Data.

For example:-names, telephone numbers, Mark, Age etc

Information: Processed data is known as information.

Database: The collection of interrelated data, usually referred to as the database.

Database-management system (DBMS): DBMS is a collection of interrelated data and a set of programs to access those data. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.

Relational Database Management system(RDBMS): DBMS in the form of related tables known as RDBMS.

Examples of RDBMS packages are

- Oracle (Oracle Corporation)
- Ingres (Computer Associates)

- SQL Server (Microsoft Corporation)
- MySQL
- Access (Microsoft Corporation)
- IMS, DB2 (IBM) and many more...

3.3.2 File System & DBMS

File System

- File system is a collection of data. Any management with the file system, user has to write the procedures
- In File system storing and retrieving of data cannot be done efficiently.
- Concurrent access to the data in the file system has many problems like
- Reading the file while other deleting some information, updating some information
- File system doesn't provide crash recovery mechanism. Eg. While we are entering some data into the file if System crashes then content of the file is lost.
- Protecting a file under file system is very difficult.

DBMS

DBMS is a collection of data and user is not required to write the procedures for managing the database.

- DBMS provides an abstract view of data that hides the details.
- DBMS is efficient to use since there are wide varieties of sophisticated techniques to store and retrieve the data.
- DBMS takes care of Concurrent access using some form of locking.
- DBMS has crash recovery mechanism, DBMS protects user from the effects of system failures.
- DBMS has a good protection mechanism.

Advantages of DBMS

Due to its centralized nature, the database system can overcome the disadvantages of the file system-based system

1. Data independency: Application program should not be exposed to details of data representation and storage. DBMS provides the abstract view that hides these details.
2. Efficient data access: DBMS utilizes a variety of sophisticated techniques to store and retrieve data efficiently.

3. Data integrity and security: Data is accessed through DBMS, it can enforce integrity constraints. E.g.: Inserting salary information for an employee.
4. Data Administration: When users share data, centralizing the data is an important task, Experience professionals can minimize data redundancy and perform fine tuning which reduces retrieval time.
5. Concurrent access and Crash recovery: DBMS schedules concurrent access to the data. DBMS protects user from the effects of system failure.
6. Reduced application development time: DBMS supports important functions that are common to many applications.

3.3.3 Data Abstraction and Data Independence

Data abstraction:-DBMS provide essential features without revealing background details. This property of DBMS known as Data abstraction.

Instances and Schemas

The collection of information stored in the database at a particular moment is called an instance of the database. The overall design of the database is called the database schema. The physical schema describes the database design at the physical level, while the logical schema describes the database design at the logical level. Database may also have several schemas at the view level, sometimes called subschemas that describe different views of the database.

DBMS Architecture and Data Independence

Here we specify architecture for database systems, called the three-schema architecture. The goal of the three-schema architecture, illustrated in Figure 3.1, is to separate the user applications and the physical database. In this architecture, schemas can be defined at the following three levels:

1. The internal level has an internal schema, which describes the physical storage structure of the database. Internal level describes how database is actually stored in the storage medium.
2. The conceptual level has a conceptual schema, which describes the structure of the whole database for a community of users. Conceptual level also known as logical level, Specify what data is actually stored in the database.
3. The external or view level includes a number of external schemas or user views. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group.

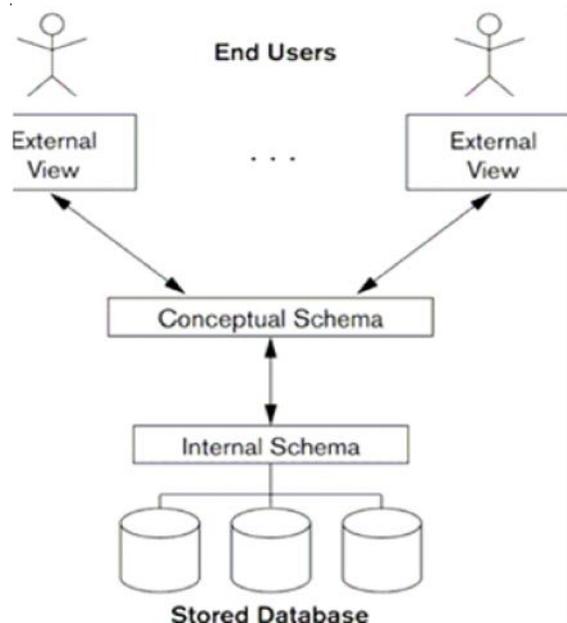


Figure 3.1

Data Independence

The capacity to change the schema definition at one level of a database system without changing the schema definition at the next higher level. We can define two types of data independence:

1. Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.
2. Physical data independence is the capacity to change the internal schema without having to change the conceptual (or external) schemas.

3.3.4 Data Model

A collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints known as data model. A database model is a specification describing how a database is structured and used.

The most common data models are

1. Relational Model

The Relational Model uses a collection of tables both data and the relationship among those data, illustrated in Figure 3.2. Each table have multiple column and each column has a unique name.

Advantages

1. The main advantage of this model is its ability to represent data in a simplified format.
2. The process of manipulating record is simplified with the use of certain key attributes used to retrieve data.
3. Representation of different types of relationship is possible with this model.

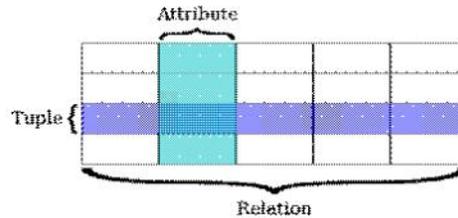


Figure 3.2

2. Network Model

The data in the network model are represented by collection of records and relationships among data are represented by links, which can be viewed as pointers, illustrated in Figure 3.3.

Advantages:

1. Representation of relationship between entities is implemented using pointers which allows the representation of arbitrary relationship.
2. Unlike the hierarchical model it is easy.
3. Data manipulation can be done easily with this model.

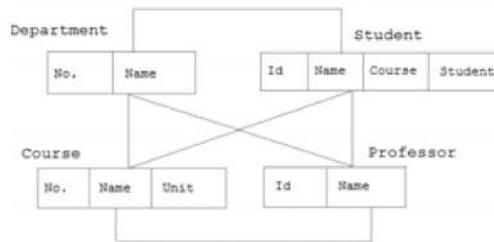


Figure 3.3

3. Hierarchical Model

A hierarchical data model is a data model which the data is organized into a tree like structure. The structure allows repeating information using parent/child relationships: each parent can have many children but each child only has one parent, illustrated in Figure 3.4. All attributes of a specific record are listed under an entity type.

Advantages:

1. The representation of records is done using an ordered tree, which is natural method of implementation of one-to-many relationships.
2. Proper ordering of the tree results in easier and faster retrieval of records.
3. Allows the use of virtual records. This result in a stable database especially when modification of the data base is made.

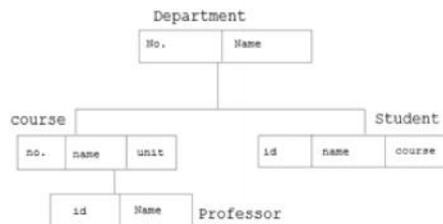


Figure 3.4

3.3.5 Relational Data Base Management System

Relation: A relation as a table with columns and rows.

Attribute: Columns of the table known as attribute.

Tuple: Rows in a table are called a tuple.

Domain: A set of atomic values that an attribute can take.

Relational Schema: The design of one table, containing the name of the table (i.e. the name of the relation), and the names of all the columns, or attributes.

Example: STUDENT (Name, SID, Age, GPA)

Degree of a Relation: The number of attributes in a relation

Cardinality: The number of rows /Records in a relation is known as cardinality

Properties of Relations

- o Relation name is distinct from all other relations
- o Each attribute has a distinct name
- o Values of an attribute are all from the same domain
- o Each tuple is distinct; there are no duplicate tuples

Data Dictionary

We can define a data dictionary as a DBMS component that stores the definition of data characteristics and relationships. You may recall that such "data about data" were labelled Metadata.

Database Users

Naive users are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.

Application programmers are computer professionals who write application programs. Application programmers can choose from many tools to develop user interfaces.

Sophisticated users interact with the system without writing programs. Instead, they form their requests in a database query language. They submit each such query to a query processor, whose function is to break down DML statements into instructions that the storage manager understands.

Database Administrator

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA). The functions of a DBA include:

- Schema definition
- Schema and physical-organization modification
- Granting of authorization for data access
- Routine maintenance

3.3.6 Keys in RDBMS

A key is an attribute or a set of attributes in a relation that identifies a tuple in a relation.

Super Keys: A super key is a set of attributes whose values can be used to uniquely identify a tuple within a relation. A relation may have more than one super key, but it always has at least one.

Candidate Keys: A candidate key is a super key that is minimal; that is, there is no proper subset that is itself a super key. A relation may have more than one candidate key, and the different candidate keys may have a different number of attributes.

Primary Key: The primary key of a relation is a candidate key especially selected to be the key for the relation. In other words, it is a choice, and there can be only one candidate key designated to be the primary key.

Alternate key: The candidate keys other than primary key is known as Alternate key

Foreign keys: The key field of one table used to establish relationship with another table known as foreign key.

Assessment Activities

Identify real life situations where the concept of DBMS is applied. Prepare a survey note (The following examples may be given as clues)

School admission register, school library register etc.

A document containing the name, age and place of the students in a class.

Practical Work

1. Consider the following table EMPLOYEE

E-Code	Name	Department	Job	Salary
1001	RAJU	EDUCATION	TEACHER	49500.00
1002	SANDHYA	EDUCATION	CLERK	25600.00
2001	SURESH	PWD	ENGINEER	56400.00
2002	REMA	PWD	PEON	14500.00
3001	HAMSA	POLICE	SHO	38900.00
4001	JOHN	EXICE	CLERK	32500.00

- Identify each of attributes
- Find degree of the above relation
- Suggest most suitable primary key
- Find the Cardinality of the above relation
- Suggest Candidate keys

TE Questions**Objective**

- Expand RDBMS
- In ----- Data model, data are organized in the form of tables
- Odd on out
(a) Oracle (b)SQL Server (c) C++ (d) ACCESS
- _____ User in DBMS is to have central control of both the data and the programs that access those data
- Choose the correct database level that is closest to the storage device.
a. External b. Logical c. Physical d. Conceptual
- In the relational model, degree is termed as _____.
a. Number of tuples. b. Number of attributes.
c. Number of tables. d. Number of constraints.
- Abstraction of the database can be viewed in _____ levels.

8. _____ model operates at the lowest level of abstraction, describing how the data is saved on storage devices.
a. External level b. Physical level c. Conceptual level d. View level.
9. In a relational model, rows are termed as _____.
a. Tuples b. Attributes c. Tables d. Cardinality
10. _____ level of data abstraction describes how the data is saved in storage devices.

Short answer

1. A bank chose a database system instead of simply storing data in conventional files. What are the merits expected by the bank?
2. Distinguish between primary key and candidate key
3. Distinguish between primary key and alternate key.
4. Describe the various levels of data abstraction. How are these levels of data abstraction related with data independence?
5. Describe logical and physical data independence?
6. Why would you choose a database system instead of simply storing data in conventional filesystem?

Essay

1. Define database management system and explain the merits and demerits of DBMS.
2. List and explain main key fields associated with DBMS
3. Explain the difference between external, internal, and conceptual schemas. How are these different schema layers related to the concepts of logical and physical data independence?

Unit - 4

SQL (STRUCTURED QUERY LANGUAGE)

Structure Query Language (SQL) is a programming language used for storing and managing data in RDBMS. SQL was the first commercial language introduced for E.F Codd's Relational model. Today almost all RDBMS uses SQL as the standard database language. SQL is a non-procedural language, that is there exist the specification about what data are needed without specifying how to get it.

Learning Outcomes

The learner

- Recognises the importance and features of Structured Query Language.
- Describe SQL Data types and commands
- Practice on SQL commands
- Identify Clauses associated with DML commands
- Perform Aggregate Functions

3.4.1 Structured Query Language (SQL)

Structured Query Language, commonly abbreviated to SQL and pronounced as "sequel", is not a conventional computer programming language in the normal sense of the phrase.

The main features of SQL:

- SQL is very simple and easy to learn.
- SQL is an ANSI and ISO standard computer language for creating and manipulating databases.
- SQL allows the user to create, update, delete, and retrieve data from a database.
- SQL works with database programs like DB2, Oracle, MS Access, Sybase, MS SQL Sever etc.

3.4.2 SQL Commands

There are three groups of commands in SQL:

1. Data Definition
2. Data Manipulation
3. Transaction Control

Data Definition Language (DDL) in SQL

The Data Definition Language (DDL) part of SQL permits database tables to be created or deleted. The most important DDL statements in SQL are:

- CREATE TABLE - creates a new database table
- ALTER TABLE - alters (changes) a database table
- DROP TABLE - deletes a database table

Data Manipulation Language (DML) in SQL

SQL language also includes syntax to update, insert, and delete records. These query and update commands together form the Data Manipulation Language (DML) part of SQL:

- INSERT INTO - inserts new data into a database table
- UPDATE - updates data in a database table
- DELETE - deletes data from a database table
- SELECT - extracts data from a database table

SQL Data Types

Numeric Data Types

Data Type	Description
SMALLINT	A small integer
INT	A standard integer
DECIMAL	A fixed-point number
FLOAT	A single-precision floating point number
DOUBLE	A double-precision floating point number

String Data Types

Data Type	Description
CHAR	A fixed-length non-binary (character) string
VARCHAR	A variable-length non-binary string
TEXT	A small non-binary string

Date and Time Data Types

Data Type	Description
DATE	A date value in 'YYYY-MM-DD' format
TIME	A time value in 'HH:MM:SS' format

SQL Integrity Constraints

SQL Integrity Constraints are the rules enforced on data columns on table. These are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the database. Following are commonly used constraints available in SQL:

- NOT NULL Constraint: Ensures that a column cannot have NULL value.
- DEFAULT Constraint: Provides a default value for a column when none is specified.
- UNIQUE Constraint: Ensures that all values in a column are different.
- PRIMARY KEY: Uniquely identified each rows/records in a database table.
- FOREIGN KEY: Uniquely identified a rows/records in any another database table.
- CHECK Constraint: The CHECK constraint ensures that all values in a column satisfy certain conditions.
- INDEX: Use to create and retrieve data from the database very quickly.

3.4.3 Practice on SQL Commands

CREATE TABLE command:

The CREATE TABLE statement is used to create a table in a database. Tables are organized into rows and columns and each table must have a name.

Syntax

```
CREATE TABLE table_name (column1 datatype [constraints], column2
datatype [constraints], column3 datatype [constraints], ..... columnN
data type [constraints] );
```

Example: To creates a new table called CUSTOMERS and adds five columns

```
SQL>CREATE TABLE CUSTOMERS (ID INT NOT NULL PRIMARY KEY ,
NAME VARCHAR (20) NOT NULL, AGE INT NOT NULL, ADDRESS CHAR
(25), SALARY DECIMAL (18, 2));
```

DROP TABLE Statement command:

The DROP TABLE statement is used to delete a table.

Syntax

```
DROP TABLE table_name;
```

Example: To remove the table structure

```
SQL>DROP TABLE CUSTOMERS;
```

ALTER TABLE command:

Syntax:

```
ALTER TABLE table_name {ADD|DROP|MODIFY} column_name
{data_type};
```

Example: To ADD a new column in an existing table:

```
SQL>ALTER TABLE CUSTOMERS ADD SEX char(1);
```

Example: To DROP sex column from existing table:

```
SQL>ALTER TABLE CUSTOMERS DROP SEX;
```

ALTER TABLE command (Rename):

The ALTER TABLE statement is used to add, delete, or modify columns in an existing table.

Syntax:

```
ALTER TABLE table_name RENAME TO new_table_name;
```

Example: To rename the above table

```
SQL>ALTER TABLE CUSTOMERS REMANE TO CUSTOMER
```

INSERT INTO command:

The INSERT INTO statement is used to insert new records in a table.

Syntax:

```
INSERT INTO table_name( column1, column2....columnN)VALUES ( value1, value2....valueN);
```

Example:Following statements would create three records in CUSTOMERS table:

```
SQL>INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (1, 'Ramesh', 32, 'Ahmedabad', 2000.00);
```

```
SQL>INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (2, 'Suresh', 25, 'Delhi', 1500.00);
```

```
SQL>INSERT INTO CUSTOMERS (ID, NAME, AGE, ADDRESS, SALARY)
VALUES (3, 'Krishna', 23, 'Kota', 2000.00);
```

All the above statements would produce the following records in CUSTOMERS table:

ID	Name	Age	Address	Salary
1	Ramesh	32	Ahmedabad	5000.00
2	Suresh	25	Delhi	1500.00
3	Krishna	23	Kota	2000.00

UPDATE command:

The UPDATE statement is used to update existing records in a table. The WHERE clause specifies which record or records that should be updated. If you omit the WHERE clause, all records will be updated.

Syntax:

```
UPDATE table_name SET column1 = value1, column2 = value2.... columnN
= valueN
[WHERE CONDITION];
```

Example: To update ADDRESS for a customer whose ID is 3?

```
SQL> UPDATE CUSTOMERS SET ADDRESS = 'Pune' WHERE ID = 3;
```

DELETE command:

The DELETE statement is used to delete rows in a table. The WHERE clause specifies which record or records that should be deleted. If you omit the WHERE clause, all records will be deleted.

Syntax:

```
UPDATE table_name SET column1 = value1, column2 = value2....
columnN=valueN
```

Example: To DELETE a customer, whose ID is 3?

```
SQL> DELETE FROM CUSTOMERS WHERE ID = 3;
```

SELECT command:

The SELECT statement is used to select data from a TABLE.

Syntax:

```
SELECT column1, column2....columnN FROM table_name;
```

Example: If you want to fetch all the fields of CUSTOMERS table, then use the following query:

```
SQL> SELECT * FROM CUSTOMERS;
```

CREATE VIEW syntax is as follows:

Syntax:

```
CREATE VIEW view_name AS SELECT column1, column2..... FROM
table_name
WHERE [condition];
```

Example: To create a view from CUSTOMERS table. This view would be used to have customer name and age from CUSTOMERS table:

```
SQL> CREATE VIEW CUSTOMERS_VIEW AS SELECT name, age FROM CUSTOMERS;
```

3.4.4 Clauses in SQL

DISTINCT Clause:

The SELECT DISTINCT statement is used to return only distinct (different) values. The WHERE clause is used to extract only those records that fulfill a specified criterion.

Syntax:

```
SELECT DISTINCT column1, column2....columnN FROM table_name;
```

Example: To produce the Salary where we do not have any duplicate entry:

```
SQL> SELECT DISTINCT SALARY FROM CUSTOMERS ORDER BY SALARY;
```

WHERE Clause:

Syntax:

```
SELECT column1, column2....columnN FROM table_name WHERE CONDITION;
```

Example: To fetch ID, Name and Salary fields from the CUSTOMERS table where salary is greater than 2000:

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY > 2000;
```

AND/OR Clause:

The AND operator displays a record if both the first condition AND the second condition are true.

The OR operator displays a record if either the first condition OR the second condition is true.

Syntax:

```
SELECT column1, column2....columnN FROM table_name WHERE CONDITION-1 {AND/OR} CONDITION-2;
```

Example: To fetch ID, Name and Salary fields from the CUSTOMERS table where salary is greater than 2000 AND age is less than 25 years:

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY
> 2000 AND age < 25;
```

BETWEEN Clauses:

The IN operator allows to specify multiple values in a WHERE clause.

The BETWEEN operator selects values within a range. The values can be numbers, text, or dates.

Syntax:

```
SELECT column1, column2....columnN FROM table_name WHERE
column_name BETWEEN val-1 AND val-2;
```

Example: To fetch ID, Name and Salary fields from the CUSTOMERS table where salary in between 2500 AND 5000:

```
SQL> SELECT ID, NAME, SALARY FROM CUSTOMERS WHERE SALARY
BETWEEN 2500 AND 5000;
```

LIKE Clause:

The LIKE operator is used to search for a specified pattern in a column.

In SQL, wildcard characters are used with the SQL LIKE operator. SQL wildcards are used to search for data within a table.

```
SELECT column1, column2....columnN FROM table_name WHERE
column_name LIKE
{ PATTERN };
```

Syntax:

Example: To display all the records from CUSTOMERS table where SALARY starts with 200:

```
SQL> SELECT * FROM CUSTOMERS WHERE SALARY LIKE '200%';
```

ORDER BY Clause:

The ORDER BY keyword is used to sort the result-set by one or more columns. This clause sorts the records in ascending order by default. To sort the records in a descending order, you can use the DESC keyword.

Syntax:

```
SELECT column1, column2....columnN FROM table_name WHERE
CONDITION
ORDER BY column_name {ASC/DESC};
```

Example: To sort the result in ascending order by NAME and SALARY:

```
SQL> SELECT * FROM CUSTOMERS ORDER BY NAME, SALARY;
```

[Default ORDER BY clause is ASC]

Example: Sort the result in descending order by NAME:

```
SQL> SELECT * FROM CUSTOMERS ORDER BY NAME DESC;
```

GROUP BY Clause:

The GROUP BY statement is used in conjunction with the aggregate functions to group the result-set by one or more columns.

Syntax:

```
SELECT SUM(column_name) FROM table_name WHERE CONDITION  
GROUP BY column_name;
```

Example: If you want to know the total amount of salary on each customer, then GROUP BY query would be as follows:

```
SQL> SELECT NAME, SUM (SALARY) FROM CUSTOMERS GROUP BY  
NAME;
```

HAVING Clause:

The HAVING clause was added to SQL because the WHERE keyword could not be used with aggregate functions.

Syntax:

```
SELECT SUM (column_name) FROM table_name WHERE CONDITION  
GROUP BY column_name HAVING (arithmetic function condition);
```

Example: To display record, which similar age count would be more than or equal to 2:

```
SQL > SELECT * FROM CUSTOMERS GROUP BY age HAVING  
COUNT(age) >= 2;
```

3.4.5 Aggregate Functions

Following is the list of all useful SQL built-in functions:

- SQL COUNT Function - The SQL COUNT aggregate function is used to count the number of rows in a database table.
- SQL MAX Function - The SQL MAX aggregate function allows us to select the highest (maximum) value for a certain column.

- SQL MIN Function - The SQL MIN aggregate function allows us to select the lowest (minimum) value for a certain column.
- SQL AVG Function - The SQL AVG aggregate function selects the average value for certain table column.
- SQL SUM Function - The SQL SUM aggregate function allows selecting the total for a numeric column.
- SQL SQRT Functions - This is used to generate a square root of a given number.

Example: To count the total number of customers

```
SQL>SELECT COUNT(*) FROM CUSTOMERS;
```

Example: To find Maximum salary

```
SQL>SELECT MAX(SALARY) FROM CUSTOMERS;
```

Example: To find Minimum salary

```
SQL>SELECT MIN(SALARY) FROM CUSTOMERS;
```

Example: To find Average salary

```
SQL>SELECT AVG(SALARY) FROM CUSTOMERS;
```

Example: To find Total salary

```
SQL>SELECT SUM(SALARY) FROM CUSTOMERS;
```

SQL SQRT Function: SQL SQRT function is used to find out the square root of any number. You can Use SELECT statement to find out square root of any number as follows:

```
SQL> select SQRT(16);
```

```
SQRT(16)
```

```
4.000000
```

Assessment Activities

- *Conduct a group discussion for the importance of SQL*
- *Class test for finding the fundamental knowledge about SQL*
- *Asks every student to create a table of their own choice. The table names can be similar, but they should have different fields.*
- *Encourage the students to enter data into the table. (at least 10 records) At this stage, ensure the difference in the structure of table, correct usage of data types and constraints in each column of table by peer to peer verification of students. Also ask the verifying student to write 10 questions based on the data on the table (questions should be framed in such a way that queries contain important commands).*
- *Then each student tries to write SQL queries for the questions. After completion of the SQL queries, peer evaluation is conducted and make necessary changes are made in the queries wherever necessary.*

Practical Activities

- I. Perform the following SQL commands
1. Create a table EMPLOYEE (E-CODE,NAME,DEPARTMENT,JOB)
 2. Alter the above table by adding a new field SALARY
 3. Insert the details of employees to create the bellow table

E-code	Name	Department	Job	Salary
1001	Raju	Education	Teacher	49500.00
1002	Sandhya	Education	Clerk	25600.00
2001	Suresh	Pwd	Engineer	56400.00

4. Rename the above table as EMPLOYEE_DETAILS
5. Increase the SALARY of SANDHYA as 27500.00
6. Display all details of Employees
7. Display different DEPARTMENT in the above table
8. Display details of all employees whose SALARY greater than 35000.00
9. Display the name of employee whose SALARY greater than 40000.00 and DEPARTMENT is PWD
10. Display JOB of all employees whose SALARY in between 40000.00 and 50000.00
11. Display the details of all employees whose name start with the letter "S"
12. Rearranging the above table in ascending order of SALARY
13. Rearrange the above table in alphabetical order of NAME
14. Write SQL command to find total SALARY of each DEPARTMENT
15. Create a VIEW from the above table containing NAME and JOB of all employees whose SALARY greater than 35000.00
16. Create a VIEW from the above table containing the details of all employees whose DEPARTMENT is EDUCATION
17. Write SQL command to find total number of employees
18. Write SQL command to find maximum SALARY in the above table
19. Write SQL command to find minimum SALARY in the above table
20. Write SQL command to find sum SALARY in the above table
21. Write SQL command to find average SALARY in the above table

II. Perform the following SQL commands

1. Create the below table STUD
2. Set the field REG. NO as PRIMARY KEY and NAME is NOT NULL

Reg. No	Name	Class	Mark

3. Insert the below details in the table

Reg. No	Name	Class	Mark
101	Ramu	10-a	450
102	Rema	10-a	360
103	Hashim	10-a	550
104	George	10-a	210
105	Sindhu	10-a	412

4. Display the details of all students
5. Display the name of student whose MARK greater than 500
6. Rearrange the above table in descending order of MARK
7. Delete the details of students whose MARK less than 400
8. Delete all the details
9. Drop the above table structure

TE Questions

Objective

1. Pick the odd one out
a) DDL b) DML c) DCL d) HLL
2. Expand DML
3. Manu wants to add a new column to a table. What type of command is used to do it?
a) DML b) DDL c) DCL d) none of the above
4. The command used to add a new column to a table.
a) Alter Table b) Create Table c) Create View d) Create column
5. The command to eliminate the table CUSTOMER from a database is:
a) REMOVE TABLE CUSTOMER b) DROPTABLE CUSTOMER
c) DELETE TABLE CUSTOMER d) UPDATE TABLE CUSTOMER
6. Which of the following is the correct order of keywords for SQLSELECT statement?
a) SELECT, FROM, WHERE b) FROM, WHERE, SELECT
c) WHERE, FROM, SELECT d) SELECT, WHERE, FROM

7. Name the Clause used to sort the details in ascending order
a) Group by b) Order by c) Like d) Distinct
8. Name the command used to remove the table structure from Database
9. Pick the odd one out.
(a) CREATE TABLE (b) SELECT
(c) UPDATE (d) INSERT
10. SQL stands for _____.

Short answer

1. What are the features of SQL?
2. (a) Classify the following SQL commands.
CREATE TABLE, INSERT INTO, ALTER TABLE, DELETE, UPDATE, DROP TABLE.
(b) List the features of each category.
3. Explain integrity constraints in SQL?
4. Advantages of SQL?
5. Explain various data types in SQL?
6. Draw a table containing 5 columns and 4 rows, Perform any three DDL and DML commands in accordance with that table

Essay

- 1) Consider the below table Employee

E-code	Name	Job	Salary
1001	Raju	Teacher	49500.00
1002	Sandhya	Clerk	25600.00
2001	Suresh	Engineer	56400.00
2002	Rema	Peon	14500.00

Write SQL commands for the following questions

- (a) Create the above table with E-CODE as primary key
 - (b) Insert the details
 - (c) Display all the details
 - (d) Display name and job of all employees whose salary greater than 45000.00
 - (e) Delete all the details
 - (f) Remove the table
2. Explain aggregate functions associated with SQL

Unit - 5

ENGINEERING GRAPHICS

Engineering Graphics is the language of Engineers. Like any other language, one should be able to READ | WRITE | SPEAK. The knowledge of Engineering Graphics is useful to both scientist as well as Engineers. Engineering graphics is a set of rules and guidelines that help to create an engineering drawing. An engineering drawing is a drawing or a set of drawings that communicates an idea, design, schematic, or model. Engineering drawings come in many forms. Each engineering field has its own type of engineering drawings. For example, electrical engineers draw circuit schematics and circuit board layouts. Civil engineers draw plans for bridges and road layouts. Mechanical engineers draw parts and assemblies that need to be manufactured. Engineering graphics teaches you how to visualize and see all sides of an object in your mind.

Learning Outcomes

The learner:

- Identifies the main application areas in Engineering Graphics
- Recognise Drawing Instruments in Engineering Graphics
- Recognise lettering, numbering and dimensioning.
- Identifies different types of lines
- Differentiate scales used in drawing
- Apply principles of Geometric construction
- Construct Basic geometric shapes
- Explain the principles of projection
- Create orthographic projection and isometric projection

3.5.1 Main application areas of Engineering Graphics are:

Engineering graphics has a well-defined set of standards by which technical drawings are produced. Major application areas of Engineering graphics are

Mechanical	: Design of machine elements, CNC machine tools, Robotics.
Automotive	: Kinematics, Hydraulics, Steering.
Electrical	: Circuit layout, Panel design, control system.
Electronics	: Schematic diagrams of PCs, ICs, etc.
Communication	: Communication network, satellite transmitting pictures, T.V Telecasting
Civil	: Mapping, contour plotting, building drawing, structural design.
Architectural	: Town planning, interior decorations, multi storied complex.
Aerospace	: Design of spacecraft, flight simulator, lofting

3.5.2 Drawing Instruments

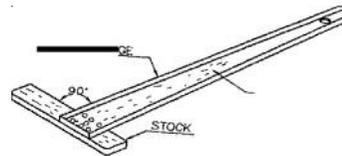
Drawing Instruments are used to prepare drawings accurately and easily. The accuracy and quality of drawing depends on the accuracy and quality of drawing instruments. The following are the commonly used materials and tools in engineering drawing.

Basic Tools and materials

- Drawing board, Drawing paper, pencil, eraser, Drawing pins/ tape, Clips, Duster.
- T-Square and Set Square, Mini drafter, Scales, Dividers and Protractor.
- Compass, French curves and Templates

T-SQUARE

T-Square helps to draw straight lines with the help of protractors and set-squares, at different angles. Since this is a tiresome and time consuming method, T-Square is an outdated instrument now. Instead we use drafting machine or mini drafter which is a combination of T-Square, Set square and protractor in one instrument.



Drafting machine

A drafting machine is a device which is mounted to the drawing board. It has rulers whose angles can be precisely adjusted with a controlling mechanism. There are two main types of apparatus: an arm-type parallelogram apparatus based on a hinged arm; and a track-type apparatus which moves on a rail mounted to the top of the drawing board. Small drafting machines (mini drafters) are commonly used.



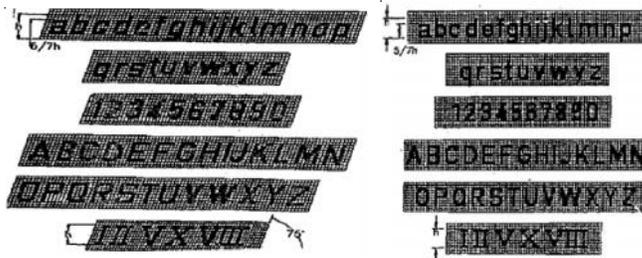
Rulers

Rulers also called Architect's scale used in technical drawing are usually made of polystyrene. It is available in two types according to the design of their edge as straight edge and grooved edge.



3.5.3 Lettering

Writing of titles, dimensions, notes and other important particulars on a drawing is called lettering. Lettering can be done in different ways such as hand lettering,



mechanical lettering etc. Mechanical lettering can be done using typewriter or computer.

Dimension

Dimension is a numerical value expressed in appropriate units of measurement and marked on a drawing with lines, symbols and notes. The dimension without any unit is considered in 'mm'. The elements of dimensioning are projection line or extension line, dimension line, leader line and arrowheads. Projection line is a thin, dark, solid line that extends from a point on the drawing to which a dimension refers. A dimension line is a thin line that shows where a measurement begins and where it ends. The dimension line should have a break in it for the dimension numbers. Dimension line should be at least 10mm from the lines of the drawing. Arrow heads are used at the ends of the dimension lines.

Points to be Considered In Dimensioning

- Each drawing shall use the same unit for all dimensions.
- Long extension line should be avoided.
- Do not cross a dimension line with another line.
- Each feature of the object shall be dimensioned only once on a drawing.
- All dimensions which are necessary to define an object or component must be clearly marked on the drawing.
- Dimension lines are placed outside the drawings except in special cases where marking inside the drawing is readable.
- In general, dimensions should be placed outside the view outline.
- Do not use a centre line or a line of the drawing as an extension or dimension line.

METHODS OF DIMENSIONS

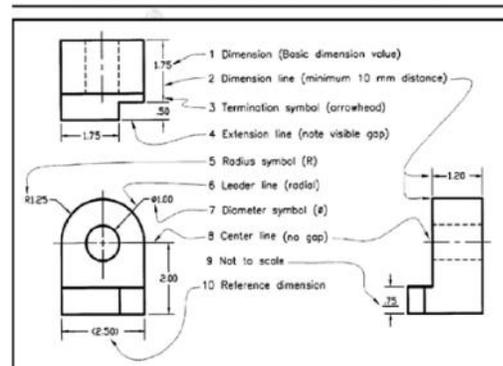
UNIDIRECTIONAL SYSTEM:

In this method dimensions shall be horizontally placed so that they can be read from the bottom of the drawing sheet. Here the dimension lines may be interrupted preferably near the middle for the insertion of dimensions.

ALIGNED SYSTEM: -

In aligned system, dimensions shall be placed parallel to and above the dimension lines, preferably in the

Dimension Basic Terminology



middle and not by interrupting the dimension lines. Here the dimensions can be read from the bottom or from the right side of the drawing.

Main components of dimensions are:

- Extension lines
- Dimension lines
- Arrow marks
- Text of dimensions
- Leader lines

FREEHAND SKETCHING

Freehand sketch is a drawing made without the help of drawing instruments. The important uses of freehand sketching are:

- It is used to convey the thoughts and ideas to the workers.
- It is used to present the ideas of the designer to the management.
- It is used for showing different layouts of the drawing.
- It is also used to convey information regarding repair or modification needed in an existing structure or machine.

COMPARISON BETWEEN FREE HAND SKETCHING AND INSTRUMENTAL DRAWING

Free hand sketching	Instrumental drawing
Drawing made without the use of drawing instrument	Drawing made with the use of drawing instrument
It is not drawn to actual scale	It is drawn to actual scale
It is not a perfect drawing	It is a perfect drawing with uniform line thickness
It is used for temporary figures/fixtures	It gives exact details of object to be manufactured

3.5.4 LINES

In engineering drawing, different types of lines are used to describe different objects. The following are some of the commonly used lines.

Types of Line	Description	General application	Pencil Grade
	Continuous thick	Object lines, Visible outlines and Visible edges.	H
	Continuous thin	Dimension lines, Projection lines, Leader lines, Extension lines, Construction lines, Section lines, and Hatching lines	2H
	Dashed thin	Hidden outline and Hidden edges	H
	Chain thin	Centre lines and Lines of symmetry	2H
	Chain thin, thick at the ends	Cutting planes	2H & H
	Continuous thin (Freehand/wavy)	Irregular boundary lines and Short break lines	2H
	Continuous thin (Zigzag)	Long break lines	2H

SCALES:

Usually the word scale is used for an instrument for drawing straight lines. But actually in Engineer's language scale means the proportion or ratio between the dimensions adopted for the drawing and the corresponding dimensions of the object.

REPRESENTING SCALES: The proportion between the drawing and the actual object can be represented by two ways as follows:

- a) Scale: - 1cm = 1m or 1cm=100cm or 1:100
- b) Representative Fraction: - (RF) = 1/100 (less than one) i.e. the ratio between the size of the drawing and the object.

There are three types of scales depending upon the proportion;

Reducing scale: When the dimensions on the drawing are smaller than the actual dimensions of the object. It is represented by the scale and RF as

Scale: - 1cm=100cm or 1:100 and by RF=1/100 (less than one)

Full scale: Sometimes the actual dimensions of the object will be adopted on the drawing then in that case it is represented by the scale and RF as

Scale: - 1cm = 1cm or 1:1 and by R.F=1/1 (equal to one).

Enlarging scale: In some cases when the objects are very small like inside parts of a wrist watch, the dimensions adopted on the drawing will be bigger than the actual dimensions of the objects then in that case it is represented by scale and RF as

Scale: - 10cm=1cm or 10:1 and by R.F= 10/1 (greater than one)

The scale or R.F of a drawing is given usually below the drawing.

The various types of scales used are

1. Plain scales
2. Diagonal scales
3. Vernier scales
4. Comparative scales
5. Scale of chords.

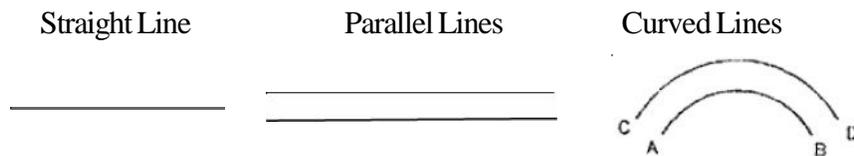
3.5.5 GEOMETRICAL CONSTRUCTIONS

It is necessary to have a good knowledge of plane geometry. The drawing of object views involves plane geometric constructions. Preparation of engineering drawings involves number of geometrical constructions. Geometrical constructions are related to straight lines, circles, arcs of circles, triangle, rectangle, square, regular polygons and conic sections.

POINT: A point represents a location in space or on a drawing and generally represented by a very small circle or a small dot.

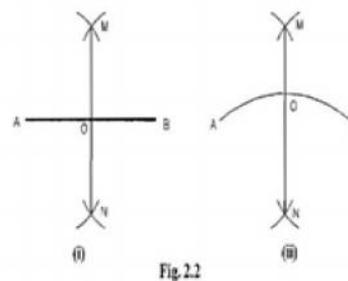
LINE: A line is the shortest distance between two points.

Types of lines:



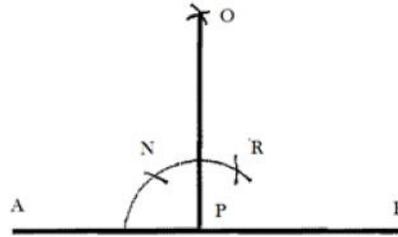
Bisect a given straight line

1. Let the given straight line or arc be AB.
2. With A and B as centers and radius greater than half of AB, draw arcs intersecting each other at M and N respectively.
3. Join M and N which bisect the given line or arc.



Drawing perpendicular

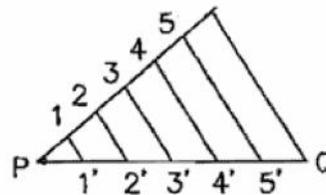
1. Let P be the given point on a given line AB.
2. With P as centre, draw an arc cutting AB at M by taking any suitable radius.
3. With same radius, mark two equal divisions on the arc MN and NR respectively.
4. With centers N and R and of any suitable radius draw arcs to intersect at a point O. Draw a line OP through O and P, the line OP is the required perpendicular line.



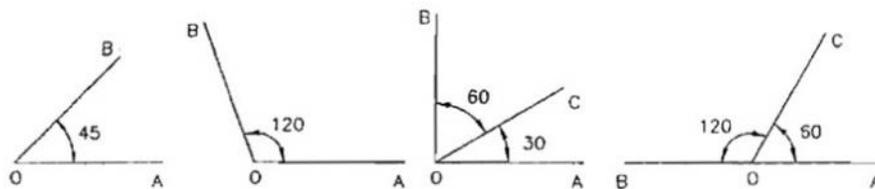
Dividing a line into equal parts

Dividing a line into number of equal parts by using dividers is not very accurate. A satisfactory method is given below.

1. If the line PQ is to be divided into six equal parts.
2. Draw a line PR inclined at any convenient acute angle to PQ.
3. Make six equal divisions along PR at any convenient length starting from P.
4. Join Q and 6. Draw lines parallel to Q6 through the division points 1, 2, 3, 4 and 5 cutting
5. PQ at 1', 2', 3', 4', and 5'.
6. Points 1', 2', 3', 4' and 5' are the division points dividing PQ into equal parts.



Angles



An angle is the inclination between two intersecting lines.

Right angle: Angle equal to 90° is called right angle.

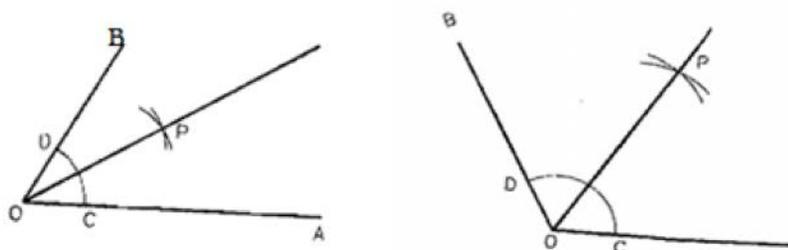
Acute angle: Angle less than 90° is called acute angle.

Obtuse angle: Angle greater than 90° is called obtuse angle.

Complementary angle: Complementary angle are those angles which together form 90° , In the fig the angles AOC and BOC are complementary angles. The angle AOC is also said to be the complement angle of BOC and vice-versa.

Supplementary angles: supplementary angles are those when two angles together make 180° .

Bisecting a given angle between two given lines:

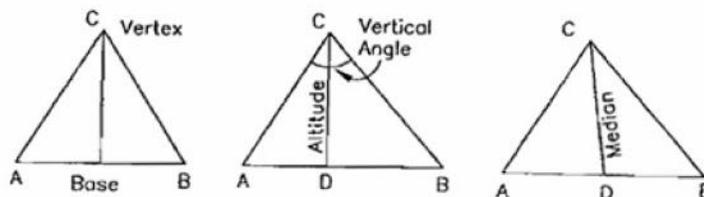


- Let the given angle be AOB between two given lines OA and OB
- With O as centre and with any convenient radius, draw an arc cutting OA at C and OB at D.
- Now with C and D as centre and at any convenient radius draw arcs to intersect each other at P. Draw a line through O and P, which bisects the given angle AO.

3.5.6 Construction of Basic Geometric Shapes

TRIANGLES:

A Triangle is a plane figure bounded by three straight lines containing three angles. The sum of the three interior angles is 180° .



- The sum of all the angles of a triangle is always 180° .
- The side on which it is supposed to stand is called its base and the angles at the base are known as base angles.
- The point where the other two sides meet is called a vertex and the angle at the vertex is called a vertical angle.

- The line drawn from the vertex and perpendicular to the base or base produced if necessary is called an altitude.
- The line joining the angular point of a triangle to the middle point of the opposite of an angular point is called the Median.

Types of triangles

Equilateral Triangle: It's that in which all the three sides are equal and the three angles are equal

Isosceles Triangle: It is that in which two sides as well as the angle opposite to them are equal

Scalene Triangle: It has no sides or angle equal. The altitude may either be within or outside the triangle.

Right angled Triangle: It is that in which one angle is equal to 90° and the side opposite to the right angle is called hypotenuse

Acute angled triangle: It is that in which all the angles are acute i.e., less than 90° .

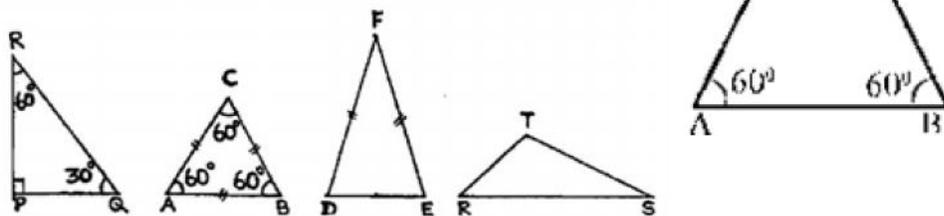
Obtuse angled Triangle: It is that in which one of its angle is obtuse and the other two angles are acute.

Drawing an equilateral triangle (given the length of one side)

Let AB be the given length of one side of an equilateral triangle.

Take 60° angle from A and draw AC

Join B and C

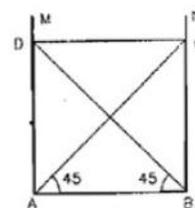


SQUARE:

Square is the quadrilateral in which all the sides are equal and the angles are at right angles. Draw a square- given the length of one side:

Let AB be the length of one side of square

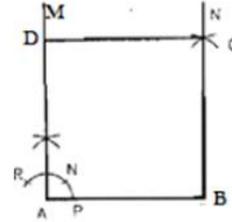
- (a) With set square only: Draw a line AB by means of a T-Square through A and B draw vertical line AM and BN.



Draw two lines AC and BD inclined at 45° to AB and BA, thereby cutting BN at C and AM at D. Join C with D. Then ABCD is the required square.

(b) With the help of compass

(i) Draw a given line AB. At A draw a line AM perpendicular to AB with A as centre and radius AB, draw an arc cutting AM at D. With B and D as centers and having same radius i.e. AB, draw arcs intersecting each other at C. Join BC and CD. Then ABCD is the required square.



(ii) Draw a straight line AB. Draw vertical lines AM and BN from A and B. With A and B as the centers draw arcs radius of AB, these arcs cut AM at C and BN at D. Join CD

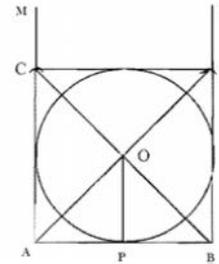
DRAW A CIRCLE INSCRIBED IN A SQUARE

Draw two diagonals AD and BC. These two diagonals intersecting at O. Draw a vertical line from O to the midpoint of line AB. Draw a circle with O as centre and radius of OP.

RECTANGLE: Rectangle is the quadrilateral in which the opposite sides are equal and all the angles are at right angles.

Procedure:

1. Draw a straight line AB equal to 50 mm
2. From A draw vertical line AM.
3. With A as centre and radius of 25 mm draw an arc cutting AM at D.
4. With D as centre and radius of 50mm draw an arc.
5. With B as centre and radius of 25mm draw another arc.
6. These 2 arcs intersect each other at C. Join BC and CD.
7. Then rectangle ABCD is obtained

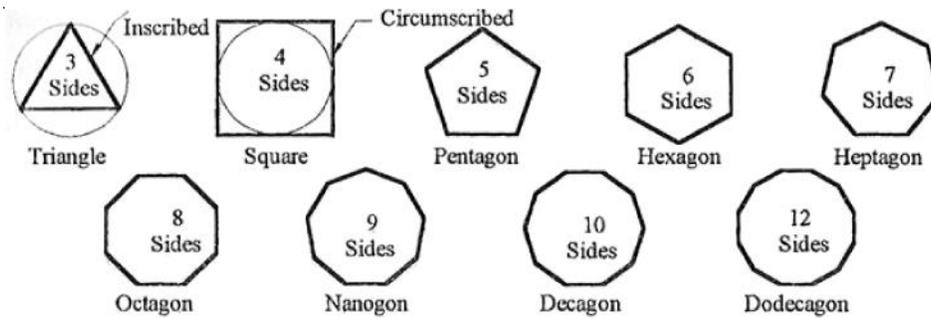


POLYGON

A polygon may be defined as a plane figure bounded by more than four straight lines and containing more than four angles.

Types of polygon: If all the sides and angles of a polygon are equal it is called a Regular polygon, but if all the sides and angles are not equal, then it is called an Irregular polygon.

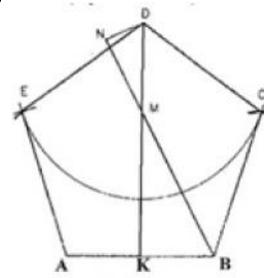
- Regular polygon can be inscribed in or circumscribed around a circle
- The polygons are named according to the number of their sides and angles.



PENTAGON

Pentagon is that which has five equal sides and angles

1. Draw a line AB equal to 35mm. bisect it at K and draw KD perpendicular to it.
2. Cut of KM=AB. Join BM and produce it to N so that MN=half of AB.
3. With D as centre and radius equal to AB. draw an arc EC.
4. With A and B as centers and having same radius cut the previous arc at E and C.
5. Join BC, CD, DE and EA. Then ABCDEF is the required Pentagon.

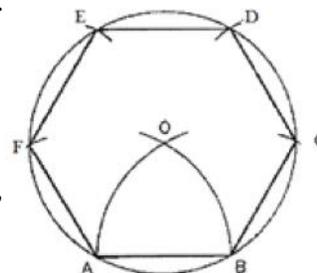


HEXAGON

Hexagon is that which has six equal sides and angles.

Draw a hexagon of given side (say 35mm side)

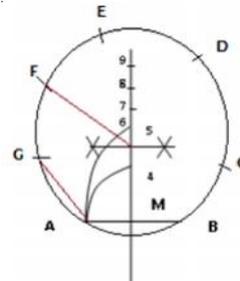
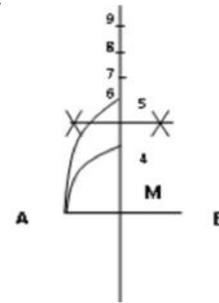
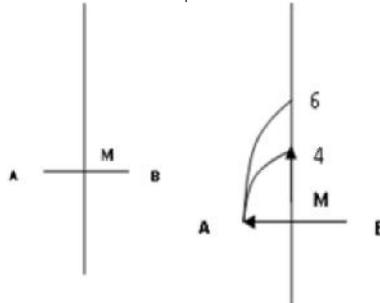
1. Draw a Line AB equal to 35mm.
2. With A and B as centers and radius of 35mm, draw arcs intersecting at O.
3. With O as the centre and having 35mm radius, draw the segment of a circle.
4. With AB as radius, cut the segment at C, D, E and F.
5. Join BC, CD, DE, EF and FA. Then ABCDEF is the required regular hexagon.



CONSTRUCT A POLYGON WITH ANY NUMBER OF SIDES

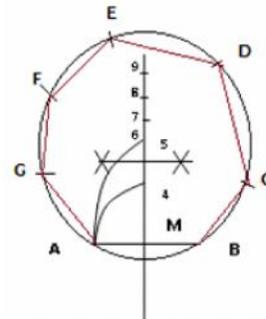
Draw one side of a polygon with side AB

1. Draw perpendicular bisector bisecting AB at M
2. Draw an arc with M as centre and MA as radius to meet the perpendicular bisector at 4.
3. Draw B as centre and BA as radius to meet the perpendicular bisector at 6.
4. Bisect 4-6 and locate the midpoint as 5.
5. Mark 7, 8 etc. on the bisector SO that $6-7=7-8=5-6$.
6. Rest of the method is explained by constructing a polygon with number of sides 7 (Heptagon).
7. Draw an arc with 7 as centre and 7-A as radius.
8. Mark the 7 corners of the heptagon on the circle by marking C, D, E etc. using a divider SO that $BC=CD=DE= \dots =AB$
9. Join AB, BC, CD, DE, EF, and FA to get the required heptagon.



TO INSCRIBE A POLYGON IN A GIVEN CIRCLE (PENTAGON)

1. Draw the circle with centre at O
2. Draw radius OA, OB, OC, etc with 72° apart
3. Draw lines AB, BC, CD, DE and EA to get the required pentagon.



The angles for different commonly used polygons are

Triangles: 120°

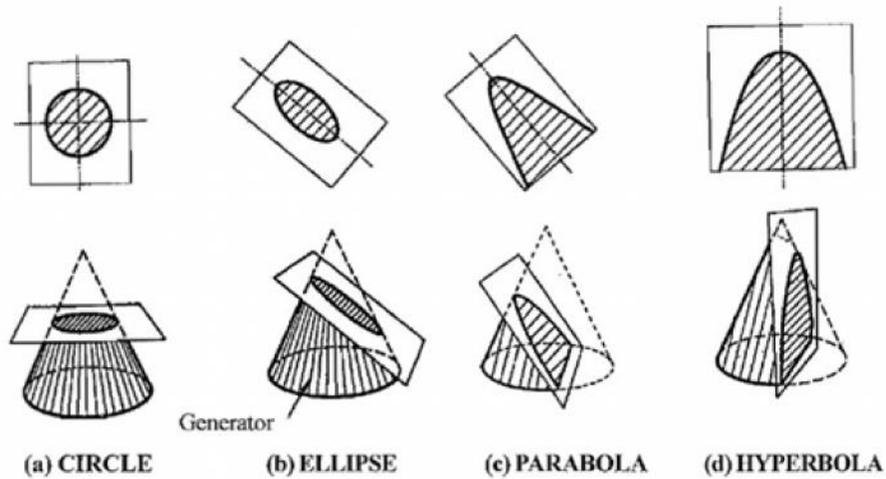
Square: 90°

Pentagon: 72°

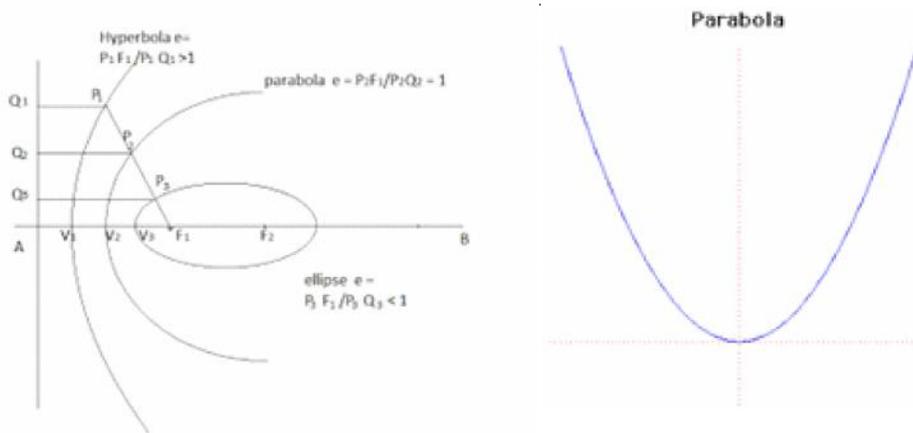
Hexagon: 60°

CONIC SECTIONS:

Conic section is the curves obtained by the intersection of a right circular cone by a plane at different angles. Ellipse, parabola, and hyperbola are the curves thus obtained and hence are called the conic sections or conics.



Conic sections are the intersections of a right regular cone, by a cutting plane in different positions, relative to the axis of the cone.



PARABOLA

The parabola is a conic section, the intersection of a right circular conical surface and a plane to a generating straight line of that surface. Given a point (the focus) and a corresponding line (the directrix) on the plane, the locus of points in that plane that are equidistant from them is a parabola.

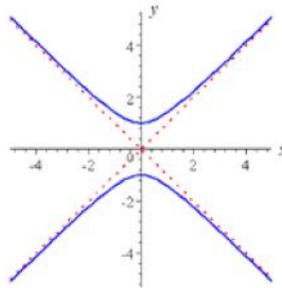
ELLIPSE

An ellipse is a plane curve that results from the intersection of a cone by a plane in a way that produces a closed curve. Circles are special cases of ellipses, obtained when the cutting plane is perpendicular to the axis. An ellipse is also the locus of all points of the plane whose distances to two fixed points add to the same constant.



HYPERBOLA

It is similar to parabola which has the eccentricity greater than 1



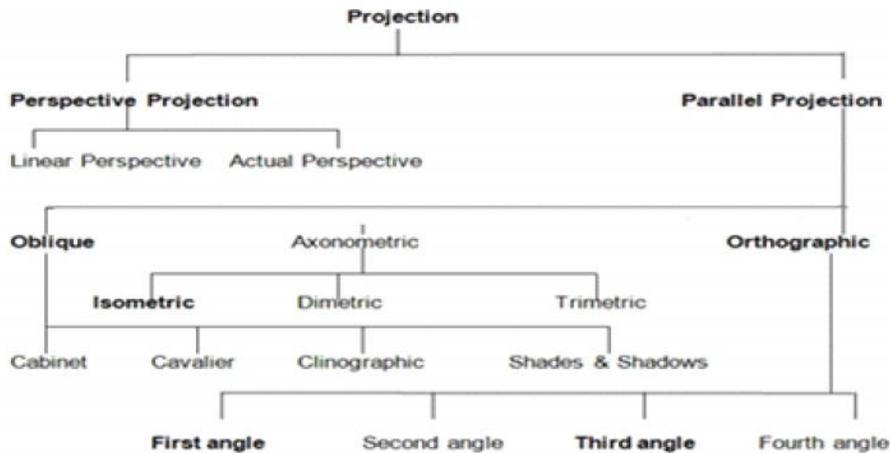
PROJECTIONS

If straight lines are drawn from various points on the contour of an object to meet a plane, the object is said to be projected on that plane. The figure formed by joining, in correct sequence, the points at which these lines meet the plane, is called the projection of the object. The lines from the object to the plane are called projectors.



TYPES OF PROJECTIONS

The projections are classified according to the method of taking the projection on the plane. A classification of projection is shown below:



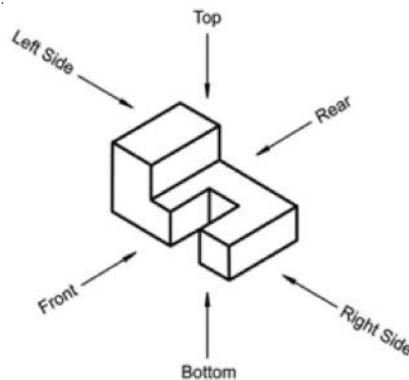
Here we discuss two important projections, orthographic projection and isometric projection.

3.5.7 ORTHOGRAPHIC PROJECTION

+ Orthographic projection is one method of projection used in engineering drawing in which the objects are projected on imaginary planes. This means we make the object become 2D. The difference between Orthographic Projection and any other drawing method is that we use several 2D views of the object instead of a single view.

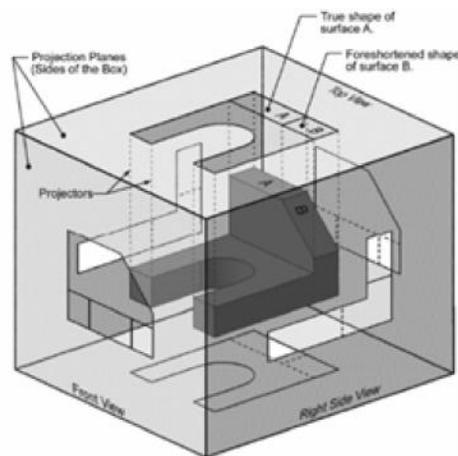
In orthographic projection the object is placed at infinite distance from the observer. The image formed on the picture plane is orthographic projection. The word orthographic means to draw at right angles.

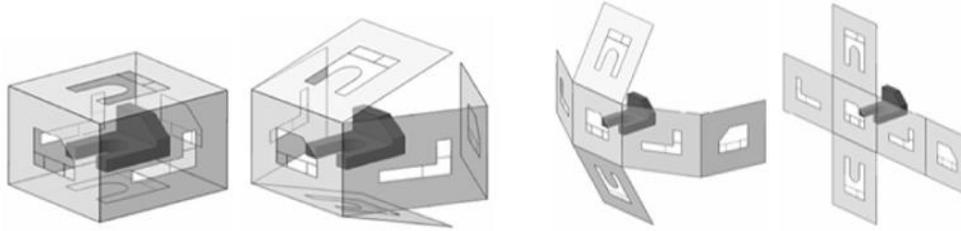
THE SIX PRINCIPLE VIEWS



THE GLASS BOX METHOD

To obtain an orthographic projection, an object is placed in an imaginary glass box. Images of the object are projected onto the sides of the box to create the six principle views. The box is then unfolded to lie flat.





CREATING AN ORTHOGRAPHIC PROJECTION

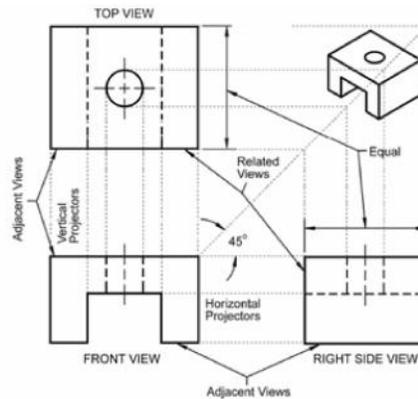
1. The following steps will take you through the creation of an orthographic projection.
2. Choose a front view. This is the view that shows the most about the object.
3. Decide how many views are needed to completely describe the object. If you are unable to determine which views will be needed, draw the standard views (front, top and right side).
4. Draw the visible features of the front view.
5. Draw projectors off of the front view horizontally and vertically in order to create the boundaries for the top and right side views.
6. Draw the top view. Use the vertical projectors to fill in the visible and hidden features.
7. Project from the top view back to the front view. Use the vertical projectors to fill in any missing visible or hidden features in the front view.
8. Draw a 45° projector off of the upper right corner of the box that encloses the front view.
9. From the top view, draw projectors over to the 45° line and down in order to create the boundaries of the right side view.
10. Draw the right side view.

Following the above mentioned steps will ensure that the orthographic projection is done correctly. That is, it will ensure that:

- The front and top views are vertically aligned.
- The front and right side views are horizontally aligned.
- Every point or feature in one view is aligned on a projector in any adjacent view (front and top, or front and right side).

- The distance between any two points of the same feature in the related views (top and right side) are equal.

Adjacent views are two adjoining views aligned by projectors. Related views are views that are adjacent to the same view.



AXONOMETRIC PROJECTION

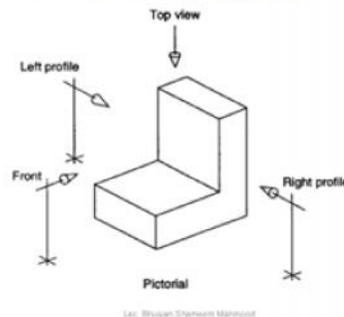
Axonometric Projection is obtained when the principal axes of the object are inclined (not perpendicular) and the line of sight is perpendicular to the plane of projection, the axonometric planes.

In most of the multi view projections, a minimum of two views are required to get an idea of the shape of the object, and even more knowledge in projection is required for better understanding of the orthographic projection. But in axonometric projection it is very easy to understand the shape of the object with a single view.

ISOMETRIC PROJECTION

Isometric projection is a method for visually representing three-dimensional objects in two dimensions in technical and engineering drawings. It is an axonometric projection in which the three coordinate axes appear equally foreshortened and the angle between any two of them is 120 degrees.

Isometric projection



Activity:

- Draw a rectangle of length 50mm and breadth 25mm. as per the procedure.
- Draw a pentagon of a given side (say 35mm side)

TE Questions:**Objective**

1. ----- lines are used to indicate imaginary features.
2. Which of the following is not a pictorial drawing?
a) isometric b) Multi view c) perspective d) axonometric
3. Hidden lines are drawn as,
(a) Dashed narrow lines (b) Dashed wide lines
(c) Long-dashed dotted wide line (d) Long-dashed double dotted wide line
4. The line connecting a view to note is called;
(a) Dimension line (b) projection line (c) leader (d) arrowheads
5. When two angles together make 90°, they are called
(a) Obtuse angle (b) Reflex angle
(c) complementary angles (d) Supplementary angles
6. The included angle of a hexagon is
(a) 30° (b) 60° (c) 120° (d) 150°
7. Name a software used for creating engineering drawings.
8. Which of the object gives a circular section, when it is cut completely by a section plane (irrespective of the angle of the section plane)?

Short Answer

1. Explain any four Basic tools in Engineering Graphics?
2. Explain the advantages of Engineering Graphics?
3. Compare free hand sketching and instrumental drawing?
4. Explain 3 types of scales used in engineering drawing.
5. Write a short note on isometric projections.
6. Write a short note on orthographic projection.

MODULE 4**WEB APPLICATION
DEVELOPMENT****Unit - 1****INTERNET AND CYBER SECURITY**

The immense growth of information and technology within a short span of time has prompted many to utilize it as a medium of communication. This idea has now been put into usage by means of Internet. This communication technology influences us on a large scale. The internet also called the Net, is the world's largest network. In this unit we identified the terminologies like Web Browser, Web Server, Web Portal, URL, WWW, Search Engines, E-Mail etc. Here unit students get an idea of Cyber security, Cybercrimes and various information technology act associated with Cybercrimes

Learning Outcomes

The learner:

- Explain the History of Internet
- Identify the terminologies Web Browser, Web Server, Web Portal and URL
- Describe Cyber ethics
- Explain Legal Issues

4.1.1 History of Internet

ARPANET (Advanced Research Projects Administration NET) is the ancestor of the Internet. The project was an experiment in reliable networking and to link defence and military research contractors introduced by Department of Defence (DOD) of US in 1969. In the 1980s, Tim Berners-Lee in the United Kingdom, introduce World Wide Web, marking the beginning of the modern Internet. Since the mid-1990s, the Internet has had a revolutionary impact on culture and commerce, including the rise of communication by electronic mail, instant messaging, voice over Internet, two-way interactive video calls, blogs, social networking, and online shopping sites.

4.1.2 Terminologies

Web browser: A web browser is a software application for retrieving, presenting and traversing information resources on the World Wide Web. An information resource may be a web page, image, video or other piece of content. Browsers read "marked up" or coded pages that reside on servers. Google Chrome, Mozilla, Microsoft Internet Explorer, Epiphany etc. are examples of Web browsers.

Web servers: A Web Server is a computer on the World Wide Web that stores web sites and related files that can be retrieved via a Web browser. Every Web server has an IP address. Web Servers come in various sizes depending on the amount of data they must handle and the expected number of daily requests for information from Internet users around the world.

Web portals: Web portals are special kind of websites which offers so many services to its users.

Eg:-www.keralagov.in

Uniform Resource Locator (URL): The address format available in internet known as URL. It is the way of naming resources and is used mainly to link pages in the World Wide Web.

E.g.- URL for VHS Examination site: <http://www.vhsexaminationkerala.gov.in>

WWW: WWW is a large storage of on line information that users can search and find with a program called web browser. It is a massive collection of web sites, all hosted on computers (called web servers) all over the world and are identified by Uniform Resource Locator (URL).

Search engines: A software that searches for information and returns sites which provide that information. The method for finding this information is usually done by maintaining an index of web resources that can be queried for the keywords or concepts entered by the user. Examples of search engines are AltaVista, Google, Yahoo etc

E-MAIL (Electronic mail): E-Mail is a system whereby a computer user can exchange messages with other computer users (or groups of users) via a communications network. Most e-mail systems today use the Internet.

Advantages of E-Mail

Fast, Easy to create, Easily accessible, Cheap, Single copy can be sent multiple recipient

Limitations of E-Mail

Can contain virus, Not legally accepted, Not secure are the limitations of E-mail

4.1.3 Cyber Ethics

Cyber security: Cyber security is the protection of information systems from theft or damage to the hardware, the software, and to the information on them as well as from disruption or misdirection of the services they provide.

Governments, military, corporations, financial institutions, hospitals and other businesses collect, process and store a great deal of confidential information on computers and transmit that data across networks to other computers. With the growing volume of cyber attacks, ongoing attention is required to protect sensitive business and personal information, as well as safeguard national security.

Network security: Network security consists of the policies adopted to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Network security involves the authorization of access to data in a network, which is controlled by the network administrator.

A network security system usually consists of many components and all components work together, minimizes maintenance and improves security. Network security components often include:

- Anti-virus and anti-spyware
- Firewall, to block unauthorized access to your network
- Intrusion prevention systems (IPS), to identify fast-spreading threats, such as zero-day or zero-hour attacks
- Virtual Private Networks (VPNs), to provide secure remote access

Cybercrime: Cybercrime, is crime that involves a computer and a network. In some cases, the computer may have been used in order to commit the crime, and in other cases, the computer may have been the target of the crime.

Types of cyber crime: Common cyber crimes are as follows:

- 1 Child Abuse or Exploitation: It involves the use of computer networks to create, distribute, or access materials that sexually exploit underage children.
- 2 Computer Intrusion/Hacking - A hacker is an unauthorized user who attempts to or gains access to an information system. It is an attack in to the privacy of data.
- 3 Cracking: It is a dreadful feeling to know that a stranger has broken into user computer systems without user's knowledge and consent and has tampered with precious confidential data and information. Cracker are differ with hacker because hacker are hired by companies to audit network security or test software but cracker do same work for their own profit or to harm others.
- 4 Credit Card Fraud (Carding): It means false ATM cards i.e. Debit and Credit cards used by criminals for their monetary benefits through withdrawing money from the victim's bank account.
- 5 E-Mail/SMS Spoofing: A spoofed E-mail/ SMS may be said to be one, which misrepresents its origin. It shows its origin to be different from which actually it originates. Here an offender steals identity of another in the form of email address, mobile phone number etc and send message via internet
- 6 Phishing: Phishing means acquire information such as usernames, passwords, credit card details, personal detail etc by electronic communication. Phishing commonly uses fake emails or fake messages which contain link of virus/ malware infected fake websites. These websites request user to enter their personal detail.

- 7 Software Piracy - It is an illegal reproduction and distribution of software for business or personal use. This is considered to be a type of infringement of copy right and a violation of a license agreement. Since the unauthorized user is not a party to the license agreement it is difficult to find out remedies.
- 8 Denial-of-service Attack: Denial-of-service referred the act by which a user of any website or service denied to the use of service of the website.
- 9 Child pornography: Which includes the creation, distribution, or accessing of materials that sexually exploit underage children

4.1.4 Legal Issues

Information technology act: The Information Technology Act, 2000 (also known as ITA-2000, or the IT Act) is an Act of the Indian Parliament notified on 17 October 2000. It is the primary law in India dealing with cybercrime and electronic commerce. The Information Technology Act, 2000 was amended in 2008.

List of offences and the corresponding penalties

Act	Crime	Punishment
65	Tampering with computer source documents	Imprisonment up to three years, or/and with fine up to Rs.200,000
66	Hacking with computer system	Imprisonment up to three years, or/and with fine up to Rs.500,000
66A	Sending offensive messages through communication service, etc.	Imprisonment up to three years, or/and with fine
66B	Receiving stolen computer or communication device	Imprisonment up to three years, or/and with fine up to Rs.100,000
66C	Using password of another person	Imprisonment up to three years, or/and with fine up to Rs.100,000
66D	Cheating using computer resource	Imprisonment up to three years, or/and with fine up to Rs.100,000
66E	Publishing private images of others	Imprisonment up to three years, or/and with fine up to Rs.200,000
66F	Acts of cyber terrorism	Imprisonment up to life.
67	Publishing information which is obscene in electronic form.	Imprisonment up to five years, or/and with fine up to Rs.1,000,000
67A	Publishing images containing sexual acts	Imprisonment up to seven years, or/and with fine up to Rs.1,000,000
67B	Publishing child porn or predating children online	Imprisonment up to five years, or/and with fine up to Rs.1,000,000 on first conviction. Imprisonment up to seven years, or/and with fine up to

		Rs.1,000,000 on second conviction.
68	Failure/refusal to comply with orders	Imprisonment up to three years, or/and with fine up to Rs.200,000
69	Failure/refusal to decrypt data	Imprisonment up to seven years and possible fine.
70	Securing access or attempting to secure access to a protected system	Imprisonment up to ten years, or/and with fine.
71	Misrepresentation	Imprisonment up to three years, or/and with fine up to Rs.100,000

ASSESSMENT ACTIVITY

- *General discussion on web portal and URL.*
- *Observe web portal and URL on internet.*
- *Conduct a debate on Cyber Crime*

TE Questions

Type: Objective

1. an information space where resources, are identified by URL
2. Name criminal activities dealing with computers and networks.

Type: Short Answer

1. Differentiate web browser and web server with example.
2. DVHSE published 2nd year result on their portal. Arun wants to know his result. How is it possible if didn't know exact web site address? Explain.
3. Explain any 3 method to protect your system from unauthorised access.
4. Match the following

A	B
a. 65	1. Sending offensive messages through communication service, etc.
b. 66A	2. Cheating using computer resource
c. 66C	3. Tampering with computer source documents
d. 66D	4. Using password of another person

Type: Essay

1. Explain any 5 cyber crimes.

Unit - 2

WEB DESIGN USING HTML

4.2.1 Basic HTML

Nowadays, Internet and World Wide Web are familiar to all, Web documents known as web pages are created using simple language HTML. HTML stands for Hyper Text Mark-up Language, developed by Tim Berners Lee in 1980. It is used by web browsers to interpret and compose text, images, and other material into visual or audible web pages. After the completion of this topic students will be able to create complicated web sites using HTML tag.

This unit describes basic HTML tags which are essential for creating static web pages. Here we introduce new terminologies like static web page and dynamic web page. HTML includes a lot of tags for web page creation such as Structure tags, common tags, line drawing tag, Page Breaking tag, formatting tags, image tags, hyperlink tags, table tags frame tag, form tag etc. After the completion of this unit students will be able to create simple web pages.

Learning Outcomes

The learner

- Explain the needs of Web pages
- Compare Static and Dynamic web pages
- Explain Tags in HTML
- List and explain Structure tags
- List and explain formatting tags
- Explain adding comments in an HTML document
- Describe Image Tags with its attributes
- Apply linking tag and its important attributes
- Classify List Tags

4.2.1.1 Why a Web Page Is Needed?

The main purpose of a web page is to share information on internet. Companies spend millions for creating brochures and distributing them. But if they have their own website customers can find out all the details online.

4.2.1.2 Static Web Page and Dynamic Web Page

A static website contains web pages which are remaining same unless their code is altered by the web master. Each page is coded in HTML and displays the same information to every visitor. They do not require any Web programming or database

design. A static site can be built by simply creating a few HTML pages and publishing them to a Web server.

Dynamic web page is a web page that displays different content each time it is viewed. Dynamic web pages are executed using a server side application program that is installed on a web server. Program code that runs on the server is called server side code. PHP, JSP, ASP etc. are used for creating dynamic web site. The web sites of Kerala State Entrance Examination, Web site of Indian railway for Ticket Reservation etc. are examples for dynamic web site.

4.2.1.3 HTML Tag

An HTML tag is a code element that tells the browser what to do with the text. It is a set of symbols defined in HTML with special meaning. Tag starts with a less than symbol (<) followed by a keyword and conclude with a greater than symbol (>).

Attributes and Values

An attribute is a keyword separated by a space within angle brackets along with the tag which provides additional information such as background colour, height, width etc. to the browser.

Consider the eg. <BODY text="red">, here text is an attribute of the tag body and it take value as red for changing the font color to red.

Container Tag and Empty Tag

Tags that requires both opening tag and closing tag are known as container tag. For eg. <html> tag is used with a pair tag </html>.

Tag that does not require closing tag is known empty tag and it doesn't span over a section.
, <HR> etc. are examples of Empty tag.

4.2.1.4 Structure Tag/Basic Tags

Structure tag is used to create the structure of an HTML document. There are four types of Structure tags.

<HTML> Tag

It is the beginning of an HTML document. The whole document is bounded with in <HTML> and </HTML> tag and is used to identify that the document is a web page.

<HEAD> Tag

Program 2.1

Simple HTML document that demonstrate structure of HTML code.

```
<html>
<head>
<title>
My first page
</title>
</head>
<body>
My First program
</body>
</html>
```

The <HEAD> tag contains information about the document.

<TITLE> Tag

This tag contains the title of the web page. The title will be visible on the title bar of the page.

<BODY> Tag

The actual contents of the web page contains in this section. This section start with <BODY> and end with </BODY>

Generating Source Code

- Open a simple text editor like notepad
- In the text editor window type the source code
- Save the file with a .htm/.html extension.

Viewing Source Code

- Open a web browser
- Open the saved file of your choice

Attributes in <Body> Tag

Body Attributes	Description
BGCOLOR	Specifies background colour for the document body. We can specify a colour name or equivalent hexadecimal number. Eg:- <BODY BGCOLOR="red">
BACKGROUND	To set an image as background for the web page. Eg:- <BODY BACKGROUND = "school.jpg">
TEXT	To specifies the font colour of the page.Eg:- <BODY TEXT="yellow">
ALIGN	Specifies the horizontal alignment of the line as LEFT, RIGHT or CENTER
LINK	Specifies the colour of the unvisited hyperlinks. Default colour is blue. Eg:- <BODY LINK = "red">
VLINK	Specifies the colour of the visited links. Standard colour is magenta Eg:- < BODY VLINK = "blue">

ALINK	Specifies the colour of the currently Active hyperlink.
	Eg:- < BODY ALINK = "yellow">
LEFTMARGIN & TOPMARGIN	LEFTMARGIN specifies the left margin from where the text in the body appears in a web page TOPMARGIN Specifies the top margin of the document window. The values are specified in pixel.

4.2.1.5 Formatting Tags

Common tags are essential for lay out of the body content.

START A NEW PARAGRAPH - <P> TAG

A paragraph can be created by enclosing text with in <P>... </P> tag.

**LINE BREAKING -
 TAG**

The purpose of this tag is to wrap the text to a new line.
 tag is an empty tag which means it has no closing tag.

CENTERING THE TEXT - <CENTER> Tag

<CENTER> </CENTER>, is container tag which used for centralising the content enclosed. The content may be text, image, table, etc.

Eg:-<CENTER> </CENTER>

<MARQUEE> Tag

This tag displays a piece of text or image scrolling horizontally or vertically in the web page. Given below is the list of important attributes that are used with <MARQUEE> tag.

Attributes	Description
HEIGHT	Sets the height of the marquee in pixels or in percentage of browser Window's height.
WIDTH	This specifies the width of the marquee in pixels or in percentage of browser window's width value.
DIRECTION	This specifies the direction in which marquee should scroll. This can have a value like up, down, left or right
SCROLLDELAY	This specifies time delay between each jump. This will have value in milliseconds like 100, 150, etc.

SCROLLAMOUNT	This specifies the speed of the marquee text
LOOP	This specifies how many times the marquee element should scroll on the screen. The default value is Infinite, which means that the Marquee scrolls endlessly.
BGCOLOR	This specifies background colour in terms of colour name or hexadecimal colour
HSPACE	This specifies horizontal space around the marquee. This can be a value in pixels or percentage value
SPACE	This specifies vertical space around the marquee. This can be a value in pixels or percentage value.

 TAG

The Tag enables to change the properties of just some part of the text without affecting the rest of the document. The default size used is 3.

Attributes	Description
FACE	Sets font of the specified font name.
SIZE	Sets size of the text. Ranges from 1 to 7.
COLOR	Sets the colour of the text.

<BASEFONT> Tag

The default font size can be changed to any value between 1 and 7 using the <BASEFONT> tag

Eg:- <BASEFONT SIZE=5>

ADD A HEADING - <Hn>..... </Hn>

HTML uses 6 levels of headings starting from H1 to H6. Level 1 is the largest and level 6 is the smallest.

<Hn>..... </Hn> Where n=1 to 6

Eg:- <H1>Vocational Higher Secondary </H1>

Text Formatting

HTML can tell a web browser to display some portions of text in italic, Bold, Superscript or even a combination. The following tags are used for formatting texts.

Bold text	-
Italicize text	- <I></I>
Strike out text	- <STRIKE> </STRIKE>
Underline text	- <U> </U>
Superscript text	- ^{.....}
Subscript text	- _{....}
Enlarge text	- <BIG> </BIG>
Reduce text	- <SMALL> </SMALL>
Making bold text	- ...

Special Characters

Sometimes it is required to use special characters in HTML documents. HTML defines symbolic codes to represent special characters. For example type for each blank space we want to add. Here are some more special characters that are useful.

Copyright	©	©
Trademark	™	™
Registered	®	® or ®
Less Than	<	<
Greater Than	>	>
Ampersand	&	&
Quote	"	"

4.2.1.6 Comments In HTML <!-- -->

HTML comments are placed within <!-- -->tag. So any content placed within <!-- --> tag will be treated as a comment and will be completely ignored by the browser.

Eg:- <!--This is my first web page --!>

Programme 2.2

HTML document that demonstrate COMMON HTML TAGS

Tiger.html

```
<!-- Save Our Environment --!>
<HTML><HEAD><TITLE>SAVE ENVIRONMENT</TITLE></HEAD>
<BODY BGCOLOR=#66FF33 TEXT=#CC3399>
<FONT COLOR = "red"><CENTER> THE ENDANGERED TIGER</
CENTER></FONT><
<P ALIGN="justify">Tigers are native to much of Asia, from some of the
coldest regions to the steamy rainforests of the Indonesian Islands. They are
```

the top predator in every ecosystem they inhabit. Until the 20th Century there were nine tiger subspecies that probably numbered over 100,000 animals. extinct-200-pound (90 kilo)Balinese .

Depending on whether there are any remaining South China tigers-nobody has seen one in years-there are either 5 or 6 tiger subspecies remaining in existence; all are endangered. All tiger subspecies put together currently amount to fewer than 3,000 endangered tigers remaining in the wild.

<HR WIDTH=100% SIZE=5 COLOR="yellow">

</BODY></HTML>

Result 2.2



4.2.1.7 Placing Image in a Web Page - Tag

Images are essential for enhancing a web page. The empty tag is used to insert images in HTML pages. The SRC attribute specifies image path.

Eg. :-

Attributes	Description
HEIGHT & WIDTH	Sets the height and width of the image display area.
ALIGN	Specifies the alignment of the image as LEFT, RIGHT, CENTER, TOP, MIDDLE or BOTTOM
ALT	This specifies alternative text to be display.
	Eg.
BORDER	Specifies border in pixel to an imageEg.

Program 2.3

HTML document that demonstrate **FORMATTING TAGS** and **IMAGE TAG**

Giraffe.html

```
<html>
<head><title>save environment</title></head>
<body bgcolor=#6699ff text=#ff6633>
<h1><font color =6666ff><center>giraffe</center></font></h1>
<marquee height=30% width=5%>save our life</marquee>
<imgsrc="giraffe.jpg" height =180 width=230 align="left"><br><br>
<font size=5><p align="left">giraffes are the <b>world's tallest
mammals</b>,
thanks to their towering legs and long necks. A giraffe's legs alone are taller
than many humans-about 6 feet (1.8 meters). These long legs allow giraffes
to <U>run as fast as 35 miles (56 kilometers)</U>an hour over short
distances and cruise comfortably at 10 miles (16 kilometers) an hour over
longer distances..</P>
<P align=justify> Giraffes use their height to good advantage and browse
on leaves and buds in treetops that few other animals can reach (acacias are
a favorite). Even the giraffe's tongue is long! The 21-inch (53-centimeter)
tongue helps them pluck tasty morsels from branches. Giraffes eat most of
the time and, like cows, regurgitate food and chew it as cud.
</P></font></body><hr width=100% size=5color="green"></html>
```

Result 2.3**4.2.1.8 Hyper Linking - <A> TAG**

In an HTML document we can create hyperlink to other documents, Image, Email, audio or video with the help of an Anchor Tag <A>. The attribute HREF specifies hyper reference. The following are some examples of hyper reference.

Inserting an e-mail link: Email Me

Open a link in a new window: Courses

4.2.1.9 List Tag

Lists are often used to present information in an easy way. There are three types of lists in HTML. They are unordered lists, ordered lists, and definition lists.

UNORDERED LIST: In Unordered list the list items are arranged with bullet mark. Unordered list is created by using container tag , . List items are specified by . The main attribute used with this tag is TYPE and can take values DISC, CIRCLE OR SQUARE.

ORDERED LIST: In Ordered list the list items are arranged with numbers of different format. Ordered list is created by using container pair , . List items are specified by . The main attributes used with this tag is TYPE and START. TYPE specifies numbering type and START specifies the starting number of the list items.

Type = A, for A, B, C.....

Type =a, for a, b, c.....

Type =I, for I, II,III.....

Type =I, for i, ii, iii.....

Type =1, for 1,2,3.....

DEFINITION LIST: Definition list is formed by a group of definitions and their description. Definition list are created by the container pair <DL>, </DL> tag. Definition term is defined by <DT>and Description is defined by <DD>.

Programme 2.4

HTML document that demonstrate List tag and Hyper Link

List.html

```
<html>
<head><title>SAVE ENVIRONMENT</title></head>
<BODY BGCOLOR=#6699FF TEXT=#CC6666>
<a href="home.html" align="right"> Back to Home </a>
<h1><FONT COLOR =660000><CENTER> We are the ENDANGERED
</CENTER></font></h1>
<IMG SRC="panda.JPG" HEIGHT =180 WIDTH=230>
<IMG SRC="REINDEER.JPG" HEIGHT =180 WIDTH=230><BR><BR>
<font size=4 COLOR="RED"><p align="left">Our 10 Most Endangered
Animals List
<OL TYPE=A>
<LI>Ivory-Billed Woodpecker
```

```

<UL><LI>A North American bird so endangered it may actually be extinct
</UL>
<LI>Amur Leopard
  <UL> <LI>The world's rarest cat: Only 40 left in Russia's Far East
  </UL>
<LI>Javan Rhinoceros
  <UL> <LI>No more than 60 of these swamp-dwelling Asian rhinos
  exist</UL>
<LI> Northern Sportive Lemur
  <UL> <LI>Here's the scarcest of Madagascar's fast-dwindling lemur
  species </UL>
<LI> Western Lowland Gorilla
  <UL> <LI>Disease and illegal hunting are wiping out this gentle
  giant</UL></OL>
<OL TYPE =I START=6>
<LI>TheSaola-An Asian Unicorn
<UL TYPE="DISC"><LI>So rare it is almost mythological, the saola hangs
  on by its hoof tips in a forest full of poachers' snares</UL>
<LI> Leatherback Sea Turtle</OL>
Six Critically Endangered Species From Our Previous Lists
<DL><DT>Kakapo Parrot
<DD>These flightless New Zealand birds are so rare they all have names
Greater Bamboo Lemur
<DT>Greater Bamboo Lemur<DD>Fewer than 100 of these Critically
  Endangered animals remain
<DT>Mountain Gorilla<DD>Their habitat is shrinking, and fewer than
  700 remain</DL>
</font> </body></html>

```

Result 2.4

We are the ENDANGERED



Our 10 Most Endangered Animals List

- A. Ivory-Billed Woodpecker
 - A North American bird so endangered it may actually be extinct
- B. Amur Leopard
 - The world's rarest cat: Only 40 left in Russia's Far East
- C. Javan Rhinoceros
 - No more than 60 of these swamp-dwelling Asian rhinos exist
- D. Northern Sportive Lemur
 - Here's the scarcest of Madagascar's fast-dwindling lemur species
- E. Western Lowland Gorilla
 - Disease and illegal hunting are wiping out this gentle giant

ASSESSMENT ACTIVITIES: 4

- *Create an attractive web page, Courses.html which contains courses in your school and short description about each course using suitable lists and other tags.*
- *Create web page activities.html which contains various club activities like NSS, Seed etc. organized in your school*
- *Create a hyperlink, Home. Clicking on Home the web main.html should be loaded from*

Practical Questions

1. Write a simple program to display web controls like button, label, textbox, image control etc
2. Design a page using checkboxes and radio buttons
3. Design an application form
4. Design a web page using tables, list boxes etc
5. Design a home page using frames
6. Write an HTML code to set an image as background for a webpage
7. Write HTML code to design paragraphs with different alignments
8. Write an HTML code to show the national pledge with line breaks
9. Write an HTML code to illustrate the text formatting tags
10. Write an HTML code to list the components used in a motherboard (in ascending order)
11. Write an HTML code to display the following list

VHSE COURSES

- | | | |
|-----------------------|-----------------------|-------------|
| 1. Engineering Stream | 2. Agriculture Stream | 3. Commerce |
| • CCT | • ABFS | • CRM |
| • CSIT | • ACHM | |
| • GDPT | | |

TE Questions**Type: Objective**

1. <P> tag is used for
2. Which heading level is usually the same size as the main text on a web page?
3. The attributes face, size and color are associated withtag.
(a)<body> (b) (c)<center> (d)<marquee>

4. Which attribute specifies an alternative text to be displayed in case were the browser is unable to display the image specified in the SRC attribute.
(a)Align (b)Height (c)Hspace (d)Alt
5. is a keyword separated by a space within angle brackets along with the tag which provides additional information.

Type: Short answer

1. Differentiate Static Web Page and Dynamic Web Page.
2. Differentiate container tag and empty tag with example.
3. Create the following web page using HTML.



4. Write attributes and their values for create a marquee with following properties.
 1. Set Height with 50 pixel
 2. The marquee element should scroll 3 times on the screen.
 3. Apply marquee background to green
5. Write HTML code to print the following special characters
i.® b. ¼ c.©
6. How can you linking an image to another page in HTML?

Type: Essay

1. Write HTML code to create the following web page.
 - **Science**
 1. Computer Science Stream
 2. Biology Stream
 3. Electronics Stream
 - **Commerce**
 4. Accountancy
 5. Computer Application
 - **Humanities**

4.2.2 ADVANCED HTML

This unit describes HTML tags which are essential for creating complicated web pages. Here introduce creation of tables in HTML with the help of table tags. After the creation of tables form and framed are introduced with the help of lab sessions. After that session learners are familiarize with frame tags which can helps us to create professional web page.

Learning Outcomes

The learner

1. Create Tables using Table Tags and its attributes
2. Design various form controls using Form Tags
3. Build multiple web pages in a single window with the help of Frame and Frameset tag
4. Generate Cascading style sheet
5. Insert Multimedia contents

4.2.2.1 <TABLE> Tag

Tables arrange the contents in the form of Rows and columns. Tables are defined with the <TABLE> tag. Tables are divided into table rows with the <TR> tag. A table row can be divided into table headings with the <TH> tag. Table rows are also divided into table data with the <TD> tag.

Attributes and description of <TABLE> tag

Attributes	Description
BORDER	The value 0 is equivalent to no border. The larger the value specified the thicker will be the border. Eg. <TABLE BORDER = 10>
BORDERCOLOR	Assigns a colour to the border
HEIGHT	Table height can be specified in pixels or as percentage
WIDTH	Table width can be specified in pixels or as percentage <TABLE BORDER = 5 WIDTH = 600 HEIGHT = 300> <TABLE BORDER = 5 WIDTH = 50% HEIGHT = 75%>
CELLSPACING	Specifies the space between cell border and table frame in pixel.

	<TABLE BORDER = 8 CELLSPACING = 10>
CELLPADDING	Specifies space between cell border and cell contents in pixel. <TABLE BORDER =8 CELLPADDING =10>
BGCOLOR	Specifies Background colour. Colour name or RGB values can be given.
ALIGN	Specified the alignment of the table. Alignment can be left, center, right.
FRAME	Frame specifies which sides of the tables outer border are visible in the browser window
BACKGROUND	Set an image to the background of a table. <TABLE BACKGROUND= "envirnment.jpg">

Attributes and description of <TH> tag

Attributes	Description
COLSPAN	To combine one or more columns <TH COLSPAN = 3 > Product list and prices </TH>
ROWSPAN	To combine one or more rows. <TH ROWSPAN=3>VHSE</TH>
HEIGHT & WIDTH	<TH WIDTH = 200 HEIGHT = 100> ITEMCODE </TH>

Attributes of <TR> tag

Attributes	Description
ALIGN	To align cell content horizontally. Alignment can be left, center, right. <TR ALIGN = right>
VALIGN	To align cell content vertically. Values are top, middle, bottom. Eg. <TR VALIGN = Top>
BGCOLOR	Background colour.Eg. <TR BGCOLOR = "#FF0000">
COLSPAN	To combine one or more columns. <TR COLSPAN = 3 > Product list and prices </TH>
ROWSPAN	To combine one or more rows. <TR ROWSPAN=3>Banner</TH>

Result 2.5**4.2.2.2 Forms**

Forms allow us to gather information from users who visit our web pages. Forms contain elements like button, text field, radio button, check boxes and combo box.

<FORM> Tag.

A form start and end with <FORM> tag. The attributes used with <FORM> tag are ACTION and METHOD.

ACTION - Where the data to get or to be posted.

Program 2.5

HTML document that demonstrate Table Tag & Hyper Link

Home.html

```

<head>
<title>Untitled 1</title>
</head>
<body>
<table BORDER=1 width=80% bgcolor=#ffff99>
<tr>
<td colspan="3"></td></tr>
<tr><td rowspan="4" style="background-color: #ffff99; vertical-align: top; padding: 5px;><h1><a href=list.html
target="support">Support these animals!</a></h1>
<h2>Click on the image for further details</h2></td>
<td><h2>Giraffe</h2></td>
<td width=100><a href="giraffe.html" target="giraffe">
</a></td></tr>

```

```

<tr><td> <h2>Tiger</h2></td><tdwidth=100><a href="tiger.html"
target="giraffe">
</td></tr>
<tr><td> <h2>Sable antelope&nbsp;</h2> </td>
<td> </td></tr>
<tr><td><h2>Wallaby</h2></td>
<td></td></tr>
</table></body></html>

```

METHOD - Method for dealing the form

The METHOD attribute may get one of the following values.

- o Get - with this method, the form data set is appended to URL specified by the action attribute with a question mark. This new URL is sent for processing.
- o Post- With this method, the form data set is included in the body of the form and sent for processing

Eg:-<FORM METHOD = "POST" ACTION = "mailto:teacher@vhse.gov.in">

<INPUT> Tag

The <INPUT> tag can be used to create a number of input controls such as text, password box, checkbox, radio button, submit button, reset button etc. The main attribute is TYPE. INPUT - Tells the computer that a form item is going to be placed here.

Type - Tells the computer what type of form item here it is a text box.

Name - is the name we assign to the box.

SIZE - denotes how many characters long this box will be.

Attribute	Values	Description
	TEXT	Create a text box for accepting character input
	PASSWORD	Create a password box. Characters will be displayed as symbols <INPUT TYPE = "password" NAME = password" SIZE = "10" MAXLENGTH = "10">
	CHECKBOX	Create a check box control and can be used for selecting more than one option from a group of option.

TYPE		<INPUT TYPE = "checkbox" NAME = "hobbies" VALUE = "PAINTING" CHECKED> PAINTING
	RADIO	Create a radio button control and can be used for selecting only one option from a group of option. <INPUT TYPE = "radio" NAME = "rating" VALUE = "Very good " CHECKED> good
	SUBMIT	Used for submission of form control
	RESET	Clear all entries and bring it to the initial state.

<TEXTAREA> Control

Text Area will allow for as many words as we want. Information such as address, which is more than single line, can be accepted through text area control.

```
<TEXTAREA ROWS =6 COLS =40> Address </TEXTAREA>
```

<SELECT> Control

The popup box provides a selection list from which the input option can be selected. The following attributes are used with <SELECT> control.

SELECT: Tells the computer another form is going here.

NAME: This is the heading of the form item.

Size: denotes the size of the box

Option selected: denotes which option will appear in the box

Option: denotes another choice that will be visible when we click on item.

```
< SELECT NAME = "AGE" SIZE = "4">
<OPTION VALUE = "child " > Under 12
<OPTON VALUE = "teen"> 13-19
<OPTION VALUE = "adult" SELECTED> 20 - 64
<OPTION VALUE = "Senior"> 65 or order
</SELECT>
```

<LABEL> Tag

<LABEL> tag provides a label for the form control. The for attribute connects the label with the input control.

```
<LABEL for=REGNO> Reg. No. </LABEL><INPUT TYPE="TEXT" ID="REGNO">
```

Eg. :- To create the following Application form,

Result 2.6

SAVE ENVIRONMENT x Contact Us x
file:///C:/Users/Documents/My%20Web%20Sites/contactus.html

Contact Form

Thank you for your interest in contacting the Wild Life Conservation Foundation.
Please use the form below to contact us at any time or see our contact details below the form.

Required fields are marked with a star: *

*First Name

*Surname

Email ID

Sex: MALE: FEMALE:

*Comment/Query:

How did you hear about us?:

4.2.2.3 Frames**FRAMESET&<FRAME> Tag**

We can create frames to divide a web browser window into sections that will display a different web page. A collection of frames in the browser window is known as a frameset.

To use frames on a page we use <frameset> tag instead of <body> tag. The <frameset> tag defines how to divide the window into frames. Each frame is indicated by <frame> tag and it defines which HTML document shall open into the frame.

Attributes of <FRAMESET> Tag

Attributes	Description
COLS	Specifies number of columns contained in the frameset and the size of each column. Width of the frame can be specified in absolute values as pixel or in % of total window width. The * can be used for variable width. Eg:<FRAMESET COLS = "50%,150,*">
ROWS	It is used to specify the rows in the frameset. For example to create two horizontal frames, use <FRAMESET ROWS="10%, 90% ">
BORDERCOLOR	Assigns a colour to the border. Colour name or code can be given.
BORDER	This attribute specifies the thickness of the border of each frame in pixels. A value of zero means no border.For example <FRAMESET BORDER ="5">

Attributes of <FRAME> Tag

Attributes	Description
SRC	Specifies the URL of the document to be loaded in the frame. Eg: <FRAME SRC = "main.html">
NAME	Gives names for the frame
TARGET	Specifies the target for hyperlink. On clicking the hyperlinks in the frame, the linked page will be loaded in the frame specified by the target attribute.
NORESIZE	Disable the resizing of the frame. Eg: <FRAME SRC="main.html" NORESIZE >
SCROLLING	Indicate whether the scrollbar is to be shown or not.

Program 2.7

HTML document that demonstrate Frame Tag

Top.html

```
<HTML><BODY BGCOLOR="green">
<IMG SRC=forest1.JPG HEIGHT=100%>
<IMG SRC=forest4.JPG HEIGHT=100%>
<IMG SRC=forest3.JPG HEIGHT=100%>
</BODY></HTML>
```

Program 2.8

HTML document that demonstrate Frame Tag

Main.html

```
<HTML>
<HEAD><TITLE>SAVE ENVIRONMENT</TITLE></HEAD>
<FRAMESET ROWS="30%,70%">
<FRAME SRC="top.html">
<FRAMESET COLS="20%,65%,*">
<FRAME SRC="left.html">
```

Program 2.9

HTML document that demonstrate Frame Tag

login.html

```
<HTML>
```

```
<BODY>
```

```
<FORM>
```

```
User ID <INPUT TYPE="text"><BR><br>
```

```
Password<INPUT TYPE="password"><br>
```

Program 2.10

HTML document that demonstrate Frame Tag

Left.html

```
<HTML><body bgcolor="green"><br><br>
```

```
<font color=red size=4>
```

```
<a href="wildlife.html" target="centre"><font color="red" size=4><b>
Wild Life </b></a><br><br>
```

```
<a href="home.html" target="window"><font color="brown"
size=4><b> Back to Nature </b></a><br><br>
```

```
<a href="list.html" target="window"><font color="brown"
size=4><b>Endangered Animals </b></a><br><br>
```

```
<a href="contact.html"><font color="brown" size=4><b> Contact Us</
b></a><br>
```

```
</font></body></HTML>
```

Program 2.11

HTML document that demonstrate Frame Tag

wildlife.html

```
<html>
```

```
<BODY BGCOLOR=#cc99cc TEXT=#336600>
```

```
<h1><FONT COLOR =006600><CENTER>.....BACK TO NATURE,
SUPPORT US..... </CENTER></font></h1>
```

```
<IMG SRC="forest2.JPG" HEIGHT =180 WIDTH=230
ALIGN="LEFT"><BR><BR>
```

```
<font size=4><p align="justify">
```

The wildlife in India comprises a mix of species of different types of organisms.

Apart from a handful of the major farm animals such as cows, buffaloes, goats, poultry, and camels, India has an amazingly wide variety of animals native to the country. It is home to Bengal tigers, Indian lions, deer, pythons, wolves, foxes, bears, crocodiles, wild dogs, monkeys, snakes, antelope species, varieties of bison and the Asian elephant. The region's rich and diverse wildlife is preserved in 120+ national parks, 18 Bio-reserves and 500+ wildlife sanctuaries across the country. India has some of the most bio diverse regions of the world and hosts three of the world's 35 biodiversity hotspots - or treasure-houses - that is the Western Ghats, the Eastern Himalayas and Indo-Burma. Since India is home to a number of rare and threatened animal species, wildlife management in the country is essential to preserve these species. India is one of the seventeen mega diverse countries.

According to one study, India along with other 16 mega diverse countries is home to about 60-70% of the world's biodiversity.

[Back](home.html)

We can use style sheets to define the formatting and layout of information on our web pages. Style sheets are also known as cascading style sheet (CSS)

4.2.2.4 Types of Style Sheets

Embedding

A style is embedded in the head section of HTML document, modifies only the document that contains it. On the web page we want to use a style sheet, type <STYLE> between the <HEAD> and </HEAD> tags

Eg. <HEAD>

<STYLE>

H1 {text-align : centre; font-style :italic}

P { color: #ff0000}

</STYLE>

</HEAD>

Linking

A style sheet external to the HTML document is linked to it through a special link in head section of the HTML file.

Eg:<head>

```
<Link REL = Style sheet TYPE = "text/css" HREF = name.css>
<head>
```

Steps

1. Create a new document in a word processor or text editor
2. Type


```
H1 { text-align: center; font-style: italic}
P {color: #FF0000}
```
3. Save the document in the text only format use the .css extension to name the document

Eg. Mystyles.css
4. On each webpage we want to use the style sheet type


```
<Link REL = Style sheet TYPE = " text/css" HREF = mystyles.css>
<HTML>
<HEAD>
<TITLE> Computer Fundamentals </TITLE>
<LINK REL = Stylesheet Type = " text/css" HREF = mystyles.css>
</HEAD>
```

INLINE

Defines the style directly in the line in which the command is specified.

Eg. <P style="color:red;"> This is red </P>

4.2.2.5 Insert Multi-media Contents <EMBED> Tag

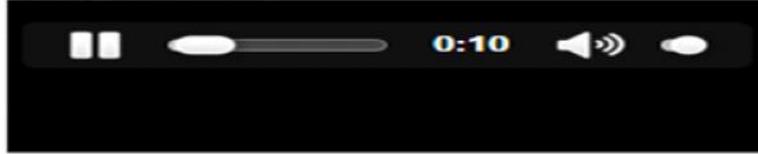
The HTML <embed> tag represents a container for external application or interactive content.

Program 2.12

HTML document that demonstrates inserting multimedia Tag

```
<Html>
<head><title>html embed tag</title></head>
<body>
<font face="garamond" size=4 color=red><b>Jungle Song<b></font><br>
<embed src="jungle.mp3" width="250" height="100"/>
</body></html>
```

Result 2.12

Jungle Song**ASSESSMENT ACTIVITY**

- List out the types of web pages available on internet developed by different scripting language (.html/.php/.asp etc.). Categorize these pages as static web page or dynamic web page with reason.
- List examples of server side script and client side script that are found on above web pages.
- Create a web page, main.html of your school. Page containing 3 or 4 paragraph about history of your school, locality, famous personalities studied and other relevant information. Page is properly aligned with the help of structure tags.

Practicals

1. Write an HTML code to create webpage contains internal marks
2. Write an HTML code to create the following table

Subject	CE	PE	TE
Physics	18	27	48
Chemistry	17	31	51
Biology	16	19	36

3. Write an HTML code to create the following table

Part I	SUBJECTS	MARKS
	ENGLISH	
Part II	ED	
	VT	
	VP	
	PP	
Part III	PHYSICS	
	CHEMISTRY	
	MATHS	

TE Questions**Type: Objective**

- Which tag is used to create a drop down list box?
a. <select> b. <list> c. <input type=list> d. <option>
- Which tag is used to insert music in your web page?

Short Answer

- Divide your window into the following partitions using frame tag.

Type: Essay

- Identify the tags, attributes and the values used to create the following web page.

Year	Free Particulars	Amount
I	Caution Deposit	100
	Tuition Fee	350
MY SCHOOL		350

- Write HTML code to create the following web page.

BIO-DATA

NAME : <input type="text"/>	
AGE : <input type="text"/>	
DATE OF BIRTH : <input type="text"/>	
SEX : <input type="radio"/> MALE <input type="radio"/> FEMALE	
FATHER'S NAME : <input type="text"/>	
MOTHER'S NAME : <input type="text"/>	
QUALIFICATION : <input type="text"/>	
ADDRESS : <input type="text"/>	
HOBBIES : <input type="checkbox"/> PAINTING <input type="checkbox"/> READING <input type="checkbox"/> LISTENING MUSIC <input type="checkbox"/> WATCHING TELEVISION <input type="checkbox"/> SINGING	
<input type="button" value="Submit"/> <input type="button" value="Reset"/>	

- What are the uses of style sheets?

Unit - 3

WEB DEVELOPMENT USING JAVASCRIPT & PHP

4.3.1 Client Side Scripting (Java Script)

JavaScript is one of the most popular client side scripting language. Now a days most of the webpage contains JavaScript. It is very easy to understand because the syntax is very similar to C++ and Java. This is an interpreter based language and source file is directly executed at runtime. JavaScript provides a large number of built-in functions for the programmers.

Learning Outcomes

The learner

- Differentiate types of scripting language
- Identifies the importance of JavaScript
- Identifies Data Types & Variables in Java Script
- Describes various operators in JavaScript
- Explain control Structures in JavaScript
- Explain functions
- Apply input form validation

4.3.1.1 Scripts

Scripts are program codes written inside HTML pages. They are written using a text editor like notepad. Scripting languages like JavaScript, VB script, PHP etc. are used for creating dynamic web pages. In an HTML page, a script is written inside <SCRIPT>and </SCRIPT>tags. It has the attributes Type and Src.

<SCRIPT Type="text/javascript" Src="testcode.js"> is used to insert JavaScript code inside the file testcode.js into an HTML file.

Different Types of Scripting Language

There are two types of scripting languages namely Server side Scripting and Client side scripting.

The client is the system on which the Web browser is running. Client side scripting is used to perform any task at the client side and is executed in the browser. The server is where the Web page and other content lives. In server side scripting, scripts are executed in server. Popular client side scripting technologies are JavaScript and VB script.

Server side scripting is a technology in which the web page containing server side scripts requested by the user is executed in the server and the result, which is an HTML code, is sent to the client browser. Server side scripting creates web pages dynamically. Popular server side scripting languages are PHP, ASP, etc.

Client side scripting and sever side scripting

| Server Side Scripting | Client Side Scripting |
|--|---|
| The Web Server executes the server side scripting. | The Web Browser executes the client side scripting that resides at the user's computer. |
| Server side scripting is used to connect to the databases that reside on the web server. | Client side scripting cannot be used to connect to the databases on the web server. |
| Server side scripting can access the file system residing at the web server. | Client side scripting can't access the file system that resides at the web server. |
| Server side scripting can't be blocked by the user. | Client side scripting is possible to be blocked by the user. |
| Response from a server-side script is slower as compared to a client-side script because the scripts are processed on the remote computer. | Response from a client-side script is faster as compared to a server-side script because the scripts are processed on the local computer. |

4.3.1.2 Importance of Javascript

JavaScript is one of the most simple, versatile and effective languages used for client side programming. An HTML page is a static one and without JavaScript it would be static still. JavaScript could make a web page dynamic and also we can create special effects on web pages.

For example, for a flash content on a web page to be executed in a web browser the JavaScript must be enabled.

JavaScript is mainly used in all websites for the validation purpose. JavaScript is not only supports the web pages but also it supports the external applications like PDF documents, running widgets, supporting for flash applications etc. JavaScript is not only limited to client side scripting but then it has steps in server side scripting too

Integrating with html

To integrate JavaScript into HTML, Netscape has extended the standard HTML tags with the SCRIPT tag. This tag helps include JavaScript in its container.

The syntax is,

```
<SCRIPT LANGUAGE= "JavaScript ">
```

```
JavaScript program
```

```
</SCRIPT>
```

JavaScript scripts can be included anywhere in the header or body of an HTML file.

In the above example, the document. write method is introduced to the user. This function is a method of the document object to print text in the client window. Another method of the document object, the document. writeln() outputs text with a carriage return added at the end of the string being displayed. This method is valid only when the <PRE>(pre format) tag of HTML is added in the file.

In the following example, the difference between writeln() and write() methods is illustrated.

Program 3.1

```
<html>
  <body>
    <pre >
      <script language="javascript" type="text/javascript">
        document. write ("Hello" );
        document.writeln("Good Morning");
        document.write ("To All" );
      </script>
    </body>
  </html>
```

Output

```
HelloGood Morning
To All
```

4.3.1.3 Data Types and Variables in Javascript

Data types: The basic data types supported by JavaScript are Number, String, Boolean and Null.

- Number contains integers and floating point values.
- String data type comprises a string of characters enclosed in double quotes.
- Boolean data type can store either true or false.
- A null data type represents nothing and has a special value in JavaScript.

Variables: Variables are storage locations used to store data. A variable is used as a reference to retrieve the value stored in its location. In JavaScript, variables are declared and used as below.

```
var example= "This is a string";
```

The syntax specifies that example is a variable that stores a string value. In JavaScript, variable names can start with a letter or an underscore (_). The remaining characters can include numbers.

4.3.1.4 Operators in Javascript

Operator	Name	Description
Arithmetic Operators		
+	Addition	Adds two operands
-	Subtraction	Subtracts the second operand from the first
*	Multiplication	Multiply both operands
/	Division	Divide the numerator by the denominator
%	Modulus	Results the remainder of an integer division
++	Increment	Increases an integer value by one
--	Decrement	Decreases an integer value by one
Comparison Operators		
==	Equal	Checks if the value of two operands are equal or not, if yes, then the condition becomes true
!=	Not Equal	Checks if the value of two operands are equal or not, if the values are not equal, then the condition becomes true.
>	Greater than	Checks if the value of the left operand is greater than the value of the right operand, if yes, then the condition becomes true.
<	Less than	Checks if the value of the left operand is less than the value of the right operand, if yes, then the condition becomes true.
>=	Greater than or Equal to	Checks if the value of the left operand is greater than or equal to the value of the right operand, if yes, then the condition becomes true
<=	Less than or Equal to	Checks if the value of the left operand is less than or equal to the value of the right operand, if yes, then the condition becomes true

Logical Operators		
&&	Logical AND	If both the operands are non-zero, then the condition becomes true.
	Logical OR	If any of the two operands are non-zero, then the condition becomes true.
!	Logical NOT	Reverses the logical state of its operand. If a condition is true, then the Logical NOT operator will make it false.
Bitwise Operators		
&	Bitwise AND	It performs a Boolean AND operation on each bit of its integer arguments.
	Bitwise OR	It performs a Boolean OR operation on each bit of its integer arguments.
^	Bitwise XOR	It performs a Boolean exclusive OR operation on each bit of its integer arguments. Exclusive OR means that either operand one is true or operand two is true, but not both.
~	Bitwise Not	It is a unary operator and operates by reversing all the bits in the operand.
<<	Left Shift	It moves all the bits in its first operand to the left by the number of places specified in the second operand. New bits are filled with zeros.
>>	Right Shift	Binary Right Shift Operator. The left operand's value is moved right by the number of bits specified by the right operand.
>>>	Right shift with Zero	This operator is just like the >> operator, except that the bits shifted in on the left are always zero.
Assignment Operators		
=	Simple Assignment	Assigns values from the right side operand to the left side operand
+=	Add and Assignment	It adds the right operand to the left operand and assigns the result to the left operand.
-=	Subtract and Assignment	It subtracts the right operand from the left operand and assigns the result to the left operand.

*=	Multiply and Assignment	It multiplies the right operand with the left operand and assigns the result to the left operand.
/=	Divide and Assignment	It divides the left operand with the right operand and assigns the result to the left operand.
%=	Modules and Assignment	It takes modulus using two operands and assigns the result to the left operand.

JavaScript evaluates conditional expressions using the ternary operator similar to the ternary operator in 'C++'. The syntax is,

(condition) ? val1 : val2

For example, the following expression assigns the value 7 to the variable A, if A > B and assigns the value 5 to A, if A < B.

(A > B) ? A = 7 : A = 5

4.3.1.5 Control Structures in Javascript

If else

It is used to execute a group of statements based on some conditions

Syntax

if(test_expression)

```
{
  Statements;
}
```

else

```
{
  Statements;
}
```

Program 3.2

Create a webpage that checks the eligibility for voting

```
<html>
  <body>
    <pre >
      <script language="javascript" type="text/javascript">
        var age=20 ;
        if (age >=18 )
        {
          document. write ("eligible for voting");
        }
        else
        {
          document. write ("not eligible for voting");
        }
      </script>
    </body>
  </html>
```

SWITCH

Switch is a multiple branching statement. Selection of different group of statements based on the value of the test expression

Syntax

```
switch ( test_expression)
{
case value1:
    Statement ;
    break;
case value2:
    Statement ;
    break;
.....
default:
    Statements;
}
```

Program 3.3

Create a webpage that performs simple arithmetic calculator

```
<html>
< HEAD > <TITLE> simple arithmetic calculator </TITLE> </HEAD>
<body>
  <pre >
    <script language="javascript" type="text/javascript">
var choice='+' ;
var a=5,b=3,result;
switch(choice)
{
  case '+':
    result=a+b;
    break;
  case '-':
    result=a-b;
    break;
  case '*':
    result=a*b;
    break;
  case '/':
    result=a/b;
    break;
  default:
    document. write ("Invalid operation");
}
document. write ("Result is :"+result);
</script>
  </pre>
</body>
</html>
```

Usage of Loops

Loops are used to repeat an action a number of times as specified in the loop construct. It is terminated based on a condition. The two types of loops are conditional and iterative loops.

The conditional loops are similar to the constructs used in other languages such as, while loops and iterative loops are those that are executed over a set range (for loops).

For and For__In Loops

The syntax of a for loop is,

for (initial value : condition : update expression)

The loop executes for the number of times specified by the condition and the update expression directs the script to either increment or decrement the initial value specified. The loop terminates once the condition is satisfied. The for-__in loops are used to step through all the properties of an object. And is used in specific applications. The syntax for a for--- in loop is,

for (variable in object){commands }

Program 3.4

This example reverses a string entered as input by the user. The function reverse() extracts the substring of the given string passed as its argument in a reversed order and displays it as output. The substring () function has two arguments firstindex and lastindex which denote the position of the string and the number of characters to be extracted.

```
<html>
<HEAD> <TITLE> String reverse </TITLE></HEAD>
<body>
  <script language="javascript" type="text/javascript">
    function reverse ( svalue ) {
      var lgth=svalue.length;
      document.write("The reversed string is") :
      for (i=lgth;i>=0;--i)
      {
        var res=svalue.substring(i,i+1) ;
        document.write(res);}
      }
      var svalue=prompt("Please enter the string");
      reverse (svalue);
    }
  </script>
</body>
</html>
```

While Loop

The while loop is a construct in JavaScript used to repeat an action until the specified condition is met. The syntax is,

While (Condition)

```
{ Javascript commands }
```

Program 3.5

```
<html>
<HEAD> <TITLE> Natural numbers </TITLE></HEAD>
<body>
<script language="javascript" type="text/javascript">
var i=1;
document.write("First 10 natural numbers are:");
while(i<=10)
{
    document.writeln(i);
    i++;
}
</script>
</body>
</html>
```

The above script is a simple illustration of the while loop, which prints the incremented value of i until the condition $i \leq 10$ is satisfied.

4.3.1.6 Functions

Function is a named part of a program which can be invoked as and when needed.

A function is defined using a function statement and is described as,

```
Functionfunction_name( arguments) { command block ;}
```

The function is always referred by the function name which is case sensitive. The function name can include underscores ($_$) and can start with a letter. The list of values to be passed to the function is enclosed in parentheses and separated by commas. The command block enclosed within curly braces defines what the function does.

Defining a function does not imply that the commands in the command block are executed. The command block is only executed when the function is called by name elsewhere in the script.

Passing Parameters

We can pass both variables and literals as arguments when calling a function. It is important to note that, if a variable is passed to a function, changing the value of the parameter within the function does not change the value of the variable passed to it.

Returning Results

Functions can return results using the return statement. The return statement can be used to return any valid expression that evaluates to a single value.

Let us now work on an example dealing with functions.

Program 3.6

```
<html>
  <head> <TITLE> Area of circle </TITLE></head>
  <body>
    <script language="javascript" type="text/javascript">
      function circle(rad)
      {
        var area;
        area=3.14*rad*rad;
        return area;
      }
      var input=prompt ("Enter the radius","1" ) ;
      var res=circle(input);
      document.write("Area of the circle is: "+ res);
    </script>
  </body>
</html>
```

Dialog Boxes

JavaScript facilitates programmers to create dialog boxes called alerts, that display a message in a small window and wait for the user to respond by clicking the OK button displayed in the box. These boxes are used to alert the user on some action. JavaScript interacts with the user by using the prompt () method. This method displays a small dialog box with a provision for text entry along with the two buttons, OK and Cancel. This method returns the text entered in the box which can be stored in a variable and used later.

Built-in Functions

Built in functions means keywords with predefined meaning. JavaScript specifications include several built-in functions and methods of base objects. Its flexibility lies in the fact that it allows programmers to create functions to supplement those already existing in JavaScript.

charAt(): Returns the character at the specified position.

Eg: Var s= "Welcome";

alert(s.charAt(3));

The output will be 'c' since the fourth character is 'c'

length(): Returns the length of the string.

```
var s= "Welcome";
alert(s.length());
```

The output will be 10 since the number of characters in the string is 10.

toLowerCase(): Returns the calling string value converted to lower case.

```
var s= "Welcome";
alert(s.toLowerCase());
```

The output will be "welcome"

toUpperCase(): Returns the calling string value converted to uppercase.

```
var s= "Welcome";
alert(s.toUpperCase());
```

The output will be "WELCOME"

substring(): Returns the characters in between two indexes in the string.

```
var s= "Welcome";
alert(s.substring(0,5));
```

The output will be "Welco"

USAGE OF EVAL () METHOD

The eval() method evaluates a numeric expression given as a string into a numeric value.

For example, the expression 10*10 can be passed as the argument to the eval() method, which in turn displays the result as 100. The syntax is,

```
eval( "10*10 " )
```

Example : The following example accepts the expression in a user prompt and evaluates the expression using the eval() method. The script illustrates the usage of the eval() method,

Program 3.7

```
<html>
<HEAD> <TITLE> Expression evaluation </TITLE></HEAD>
  <body>
    <script language="javascript" type="text/javascript">
      var quest = prompt ( "Enter the expression to evaluate" ) ;
      document. write ("Result is :" + eval (quest ) ) ;
    </script>
  </body>
</html>
```

Output

As in other programming languages, in JavaScript, functions are defined by name and invoked using this name. These functions can accept arguments and return results, similar to the methods of base objects. For example, `prompt()` and `confirm()` are two methods that operate this way.

4.3.1.7 Input form Validation

Here we have a form 'myForm' and needs to be validated. For that a javascript function called 'validateForm' is written and assigned to the `onsubmit` event handler of the 'myForm'. When a user hits the login button, the javascript code runs and checks if both the input fields, `ieusername` and `password` is empty or not.

```
<form name="myForm" onsubmit="return validateForm()" method="post">
Username: <input type="text" name="username">
Password: <input type="password" name="password">
<input type="submit" value="Login">
</form>
<script type="text/javascript">
function validateForm() {
    var x = document.forms["myForm"]["username"].value;
    var y = document.forms["myForm"]["password"].value;
    if (x == null || x == "") {
        alert("Name must be filled out");
        return false;
    }
    if (y == null || y == "") {
        alert("Password must be filled out");
        return false;
    }
}
</script>
```

ASSESSMENT ACTIVITY

- Create a table `events.html` which contains various events and description.
- Create a hyperlink, `Home`. Clicking on `Homemain.html` should be loaded.

Practical Questions

1. Design a login page using javascript
2. Find perimeter of a circle using javascript

TE Questions**Type: Objective**

1. Which of the following is correct about JavaScript?
 - a) JavaScript is a lightweight, interpreted programming language.
 - b) JavaScript has object-oriented capabilities that allow you to build interactivity into otherwise static HTML pages.
 - c) The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.
 - d) All of the above.
2. Javascript is _____ language.
 - a) Application b) Programming c) Scripting d) None of these
3. JavaScript is _____ Side Scripting Language.
 - a) Browser b) Server c) ISP d) None of These
4. Which of the following type of variable is visible only within a function where it is defined?
 - a) Global variable b) local variable c) Both of the above.
 - d) None of the above.
5. JavaScript is designed for following purpose
 - a) To Style HTML Pages b) To add interactivity to HTML Pages
 - c) To Execute Query Related to DB on Server.
 - d) To Perform Server Side Scripting Operation
6. What should appear at the very end of your JavaScript?
 - a) `</script>` b) `<script>` c) END statement. d) None of the above.
7. _____ tag is an extension to HTML that can enclose any number of JavaScript statements.
 - a) `<SCRIPT>` b) `<BODY>` c) `<HEAD>`. d) `<TITLE>`.
8. What is the correct JavaScript syntax to write "Hello World"?
 - a) `System.out.println("Hello World")` b) `println("Hello World")`
 - c) `document.write("Hello World")`. d) `response.write("Hello World")`.

9. Which of the following event fires when the form element loses the focus:

`<button>`, `<input>`, `<label>`, `<select>`, `<textarea>`?

a) onfocus b) onblur c) onclick. d) ondblclick.

10. What will be the output of the following program

```
<script type="text/javascript">
```

```
x=4+"4";
```

```
document.write(x);
```

```
</script>
```

a) 44 b) 8 c) 4. d) Error.

11. What will be the output of the following program

```
<script language="javascript">
```

```
function x()
```

```
{ document.write(2+5+"8"); }
```

```
</script>
```

a) 258 b) Error c) 7. d) 78.

12. What will be result of following javascriptfunction

```
<script type="text/javascript">
```

```
var name = "HelloWorld";
```

```
functionDisplayName () {
```

```
var name = "Hai";
```

```
document.write(name); }
```

```
</script>
```

a) HelloWorld b) Hai c) Error. d) None of above

13. What will be result of following javascriptfunction

```
<script type="text/javascript">
```

```
var name1 = "javascript";
```

```
functionDisplayName () {
```

```
var name2 = " HTML";
```

```
document.write(name1+name2); }
```

```
</script>
```

a) javascript HTML b) javascript HTML

c) Object required error. d) Javascript Error

14. What will be result of following javascript function?

```
<script type="text/javascript">
```

```

var name1 = "javascript";
functionDisplayName () {
var name2 = "HTML";
if(name1 != "")
document.write(name1);
else
document.write(name2);}
</script>

```

- a) Javascript Error at else statement b) HTML c) javascript.
d) Javascript Error in if condition.

15. What will be result of following java script function?

```

<script type="text/javascript">
var name1 = "javascript";
functionDisplayName () {
var name2 = "HTML";
(name1 != ""?document.write(name2):document.write(name1));}
</script>

```

- a) javascript b) HTML c) Javascript Error near conditional statement.
d) Nothing will happen

16. Which built-in method returns the calling string value converted to lower case?

- a) toLowerCase() b) toLower() c) changeCase(case)
d) None of the above.

Type: Short Answer

1. Explain the difference between write (^) and writeln () with the help of example.
2. What are the basic datatypes in Java Script?
3. What are the different comparison operators in Java Scri?
4. Write a Javascript program to find the given number is odd or even.
5. Explain the difference between for and for_inloops.

Type: Essay

1. Compare client side scripting and server side scripting languages.
2. Write javascript program to display the multiplication table of a given number

3.2 SERVER SIDE SCRIPTING USING PHP

This unit is an introduction to the PHP-Hypertext Pre-processor, server side scripting language. You may have visited some interactive web site where the URL ends with a file extension of .html, .php, .php3, .php4 or .php5. PHP can be used to create interactive dynamic websites and rapidly gaining in popularity because it is flexible, cross platform technology that provides amazingly powerful features. Now we are familiar with static web developing with the help of HTML and JavaScript. This unit introduces powerful concept of dynamic web site development.

Learning Outcomes

- Explain the features of PHP
- Explain syntax of PHP statement
- Describe adding comments PHP documents
- Differentiate variables and constant in PHP
- Identify various data types and operators used in PHP
- Explain control structures in PHP
- Explain functions in PHP
- Understand the concepts of Arrays
- Design PHP Forms
- Understand PHP global variables and super global arrays
- Explain database functions used in PHP
- Establish connection to databases (MySQL) using PHP database functions

4.3.2.1 Features of PHP

PHP stands for "PHP Hyper Text Preprocessor". It was developed by Rasmus Lerdorf in 1994 and PHP is known at that time as Personal Home Page.

PHP is a programming language for building dynamic interactive websites. PHP is a cross platform, html embedded, and server side web scripting language.

PHP file extensions - .phtml, .php, .php3 or .php4

Important features of PHP are given below

- PHP is open source program to create dynamic web pages
- PHP runs on various platforms (Windows, Linux, UNIX, Mac OS X, etc.)

- PHP is easy to learn and runs efficiently on all servers used today (Apache, IIS, etc.)
- PHP supports a wide range of databases
- PHP is Cross platform
- HTML embedded: PHP codes are written in files containing a mixture of PHP and html codes.
- PHP provides many levels of security

Program 3.8

Simple PHP program

```
<html>
body>
<?php
echo "Hello!!! World!!!";
?>
</body>
</html>
```

4.3.2.2 Basic PHP Syntax

A PHP script can be placed anywhere in the document. A PHP script starts with <?php and ends with ?>. PHP is not case sensitive. Put semicolon (;) at the end of each PHP statement. The built-in PHP function "echo" or "print" is used to output the text on a web page.

4.3.2.3 Comments In PHP

A comment in PHP code is a line that is not read/executed as part of the program. Its only purpose is to be read by someone who is looking at the code. Comments can be used to let understand others what you are doing. Comments can be written as follows.

- Any text between // and the end of that line is a single line comment
- Single line comments may begin with a # character
- Any text on one or more lines between /* and */ is ignored

4.3.2.4 Variables and Constants

Variables: Variables are storage location for holding data. The content of a variable can be changed during the execution of a program.

Variable naming conventions

- All variables in PHP starts with a dollar (\$) sign.
- PHP variables may not be begun with a digit.

Program 3.9

Demonstration of comment

```
<html>
<body>
<?php
// This is a comment and is not read
# This is also a single line comment
/* This is also a comment
   But span more than one line */
?>
</body>
```

- When naming variables any letter, digit and the underscore (_) character may be used
- The variable statement should end with a semicolon terminator
- Variables with more than one word should be separated with underscores.
- Variables with more than one word can also be distinguished with capitalization. \$myVariable

Program 3.10

Demonstration of declaration of variables

```
<?php
$num = 10;
$str = 'VHSE';
$fl= 6.8;
print $fl; echo $num; echo $str;
?>
```

PHP has three different variable scopes:

- Local - A variable declared within a function has a LOCAL SCOPE and can only be accessed within that function:
- Global - A variable declared outside a function has a GLOBAL SCOPE and can only be accessed outside a function
- Static - Normally, when a function is completed/executed, all of its variables are deleted. However, sometimes we want a local variable NOT to be deleted. We need it for a further job. To do this, use the static keyword when you first declare the variable.

Constants: Identifier values do not change is called constant. The value cannot be changed during the script.

- No \$ sign is used in a constant
- To define a constant use define function.

Program 3.11

Demonstration of declaration of Constant

```
<?php
define ("Pi", 3.14);
define ("name", "Varun");
echo name;
print Pi;    ?>
```

4.3.2.5 Data Types and Operators

Data types: Variables can store data of different types, In PHP commonly used data are Integer, Double, Boolean, String, Object, Array, Null, Resource

Operators : There are many operators used in PHP, so we have separated them into the following categories to make it easier to learn.

1. Assignment operators

Assignment operators are used to set a variable equal to a value or set a variable to another variable's value. Such an assignment of value is done with the "=", or equal character. Example:

```
$my_var = 4;
```

2. Arithmetic operators

The PHP arithmetic operators are used with numeric values to perform common arithmetical operations, such as addition, subtraction, multiplication etc.

Operator	Meaning	Example
+	Addition	2 + 4
-	Subtraction	5 - 3
*	Multiplication	5 * 3
/	Division	15 / 3
%	Modulus	43 % 10
++	Increment	
--	Decrement	

3. Comparison operators

Comparisons are used to check the relationship between variables and/or values. Result of comparison will always be true or false.

Operator	Meaning	Example	Result
==	Equal To	\$x == \$y	FALSE
!=<>	Not Equal To	\$x != \$y	TRUE
<	Less Than	\$x < \$y	TRUE
>	Greater Than	\$x > \$y	FALSE
<=	Less Than or Equal To	\$x <= \$y	TRUE
>=	Greater Than or Equal to	\$x >= \$y	FALSE

4. Logical operators

Operator	Meaning	Example	Result
&&	Logical AND	<code>\$x&&\$y;</code>	True if both x and y are true
And	Logical AND	<code>\$x and \$y;</code>	True if both x and y are true
	Logical OR	<code>\$x \$y;</code>	True if either x or y are true
OR	Logical OR	<code>\$x OR \$y;</code>	True if either x or y are true
XOR	Logical Exclusive XOR	<code>\$x XOR \$y;</code>	True if either \$x or \$y is true, but not both
!	Logical Not	<code>!\$x;</code>	True if \$x is not true

5. String operators

The period "." is used to add two strings together, the period is the concatenation operator for strings.

Program 3.12

Demonstration of string operator

```
<?php
$a= "Vocational";
$b="Higher Secondary";
echo $a.$b; ?>
```

6. Combination arithmetic & assignment operators

Operator	Meaning	Example	Result
<code>+=</code>	Plus Equals	<code>\$x += 2;</code>	<code>\$x = \$x + 2;</code>
<code>-=</code>	Minus Equals	<code>\$x -= 4; \$</code>	<code>x = \$x - 4;</code>
<code>*=</code>	Multiply Equals	<code>\$x *= 3;</code>	<code>\$x = \$x * 3;</code>
<code>/=</code>	Divide Equals	<code>\$x /= 2;</code>	<code>\$x = \$x / 2;</code>
<code>%=</code>	Modulo Equals	<code>\$x %= 5;</code>	<code>\$x = \$x % 5;</code>
<code>.=</code>	Concatenate Equals	<code>\$str="hello";</code>	<code>\$str= \$str . "hello";</code>

7. Conditional operator

(test expression)? Statement to be executed if condition is true: Statement to be executed if condition is false

4.3.2.6 Control Structures

Control structures are statements that alter the normal execution sequence of a program.

Decision Making Statements

A decision statement contains an expression, which is evaluated first. Depending on the value of the expression program statement blocks are selectively executed. In PHP we have the following conditional statements:

- if statement - statement block if one condition is true
- if...else statement - statement block if condition is true and another block of statement if that condition is false
- if...elseif...else statement - executes different codes for more than two conditions
- switch statement - selects one of many blocks of code to be executed

If Statement

The if statement executes some code if one condition is true.

Syntax

```
if (condition)
{
code to be executed if condition is
true;
}
```

if...else statement

The if...else statement executes some code if a condition is true and another code if that condition is false.

Syntax

```
if (condition){
code to be executed if condition
is true;
}
else {
code to be executed if condition is false;
}
```

Program 3.13

Demonstration of if statement

```
<?php
$a=10; $b=20;
if ($a > $b)
    echo $a. "is the largest";
if ($b > $a)
    echo $b. "is the largest";
?>
```

Program 3.14

Demonstration of if statement

```
<?php
$a=10; $b=20;
if ($a > $b)
    echo $a. "is the largest";
else
    echo $b. "is the largest";
?>
```

if...elseif...else statement

The if...else if...else statement executes different codes for more than two conditions

Syntax

```
if (condition)
{
code to be executed if this condition
is true;}
elseif (condition) {
code to be executed if this condition
is true;}
elseif (condition) {
code to be executed if this condition
is true;}
else
{
code to be executed if all
conditions are false;
}
```

SWITCH ...CASE Statement

Use the switch statement to select one of many blocks of code to be executed. Several equality conditions are checked and statement blocks are executed according to the values.

```
switch (n) {
case label1:
code to be executed if n=label1;
break;
case label2:
code to be executed if n=label2;
break;
```

Program 3.15

Demonstration of if ...elseif...else statement

```
<?php
$Day=date("D");
if ($Day=='Sun')
echo"Have a Nice Sunday";
elseif($Day=='Fri')
echo"Have a Nice Weekend";
else
echo"Have a Nice Day";
?>
```

Program 3.16

Demonstration of switch ...case statement

```
<?php
switch($Day)
{case"Sun":
echo"Have a Nice Sunday";
break;
case"Fri":
echo"Have a Nice Weekend";
break;
default:
echo"Have a Nice Day";
}?>
```

Program 3.17

Program to print natural numbers from 1 to 10

```
<?php
$I=1;
echo 'Natural numbers from 1 to 10';
echo "<br>";
while($I<=10)
{echo $I;
echo "<br>";
$I=$I+1;
}
?>
```

```

    case label3:
code to be executed if n=label3;
        break;
    ...
    default:
code to be executed if n is different from all labels;}

```

Looping Statement

Loop statements are used for executing a group of statements repeatedly. Loop statement reduces the size of the code. In PHP there are mainly 4 types of loops.

While loop

While loop execute a block of statement based on a condition. The condition is evaluated first and execution follows till the condition is true.

Syntax

```

While (condition)
{
Code to be executed;
}

```

Do...while Loop

The do...while loop will always execute the block of code once, it will then check the condition, and repeat the loop while the specified condition is true.

Syntax

```

do{
code to be executed while condition
is true;
} while (condition);

```

Program 3.18

HTML code for inputting number to find factorial

```

<html>
<body>
<form method="post" action="fact.php">
Enter the Number <input type="text"
name="num">
<br>
<input type="Submit">

```

Program 3.19

PHP code for inputting number to find factorial

```

<?php
$num=$_POST[num];
$fact=1;
for($I=1; $I<=$n; ++$I)
{
$fact=$fact*$I;
}
$I--;
echo 'Factorial of '.$I. ' is '.$fact;
?>

```

For loop

The for loop is used when you know in advance how many times the script should run.

Syntax

```
for (initialisation; test condition; increment)
{
    code to be executed while condition is true;
}
```

4.3.2.7 Functions

A function is a block of statements that can be used repeatedly in a program. In PHP there are usually two types of functions. Besides the built-in functions, we can create our own user defined functions.

Built-in-functions

Built in functions are the functions that are provided by php library. Many activities are carried out using library functions. These functions perform file access, mathematical computations, string functions, date functions etc.

String Functions

- strtoupper() : Converts to uppercase.
- strtolower() : Converts to lowercase.
- strlen() : Returns the length of the string.
- strrev() : reverse of the string.
- ucfirst() : Convert first character to uppercase.
- ucwords() : it returns the given string as title case
- ltrim () : it returns the string by trimming the left side spaces
- rtrim () : it returns the string by trimming the right side spaces

User Defined Function

User defined function declaration starts with the word "function". User defined function can be created as follows;

Syntax

```
function functionName()
{
    code to be executed;
}
```

Function Arguments

Information can be passed to functions through arguments. An argument is just like a variable. Arguments are specified after the function name, inside the parentheses. You can add as many arguments as you want, just separate them with a comma.

PHP Functions - Returning Values

To let a function return a value, use the return statement:

4.3.2.8 Arrays

An array is a special variable, which can hold more than one value at a time. In PHP, the array() function is used to create an array. There are three types of arrays in PHP. They are:

- Indexed arrays - Arrays with a numeric index
- Associative arrays - Arrays with named keys
- Multidimensional arrays - Arrays containing one or more arrays

Indexed Arrays

There are two ways to create indexed arrays. The index can be assigned automatically (index always starts at 0), like this:

```
$colours = array("Red", "Blue", "Green");
```

or the index can be assigned manually:

Program 3.20

PHP program to check whether the given number is odd or even using function

```
<?php
functionis_odd()
{
    global $num;
    echo("$num is an odd number<hr>");}
functionis_even()
{
    global $num;
    echo("$num is an even
number<hr>");}
?>
<html><body>
<?php
    $num=57;
    ($num % 2 !=0) ? is_odd() : is_even();
    $num=44;
    ($num % 2 !=0) ? is_odd() : is_even();
?>
</body></html>
?>
```

Program 3.21

PHP program to check whether the given number is odd or even using function with argument

```
<?php
functionis_odd($num)
{
    echo("$num is an odd number<hr>");}
functionis_even($num)
{
    echo("$num is an even
number<hr>");}
?>
<html>
</php
($num % 2 !=0) ? is_odd(57) : is_even(57);
    ($num % 2 !=0) ? is_odd(44) :
is_even(44);
?>
</html>
```

```
$ colours [0] = "Red";$ colours [1] = "Blue";$ colours [2] = "Green";
```

ASSOCIATIVE ARRAYS

Associative arrays are arrays that use named keys that you assign to them. There are two ways to create an associative array:

```
$RegNo= array("Akhil"=>"111351","George"=>"111352", "Seetha"=>"111357");
```

Or

```
$RegNo['Akhil'] = "111351";
```

```
$RegNo['George'] = "111352";
```

```
$RegNo['Seetha'] = "111357";
```

4.3.2.9 PHP Forms

You know that forms allow us to gather information from users who visit our web pages. Examples of form elements are text box, password, text areas, radio button, check boxes etc. which allow users to enter information.

FORM Tag.

```
Eg:-<FORM METHOD="POST"
ACTION = "mailto:
vhss903002@gmail.com">
```

<FORM> - Told the computer a form was starting.

METHOD - Method for dealing the form

<ACTION> - Where the data to get or to be posted.

The method attribute may get one of the following values.

1. Get- with this method, the form data set is appended to URL specified by the action attribute with a question mark. This new URL is sent for processing. GET requests have length restrictions. Information sent from a form with the GET method is visible to everyone (all variable names and values are displayed in the URL).

Program 3.22

PHP program for form

```
<HTML>
<BODY>
<FORM>
  User Name <INPUT TYPE="text"><BR>
           Password <INPUT
TYPE="password"><BR>
<INPUT      TYPE="submit"
VALUE="SUBMIT">&nbsp;
&nbsp; <INPUT  TYPE="reset"
VALUE="RESET" >
</FORM>
</BODY>
</HTML>
```

2. Post- With this method, the form data set is included in the body of the form and sent for processing. Information sent from a form with the POST method is invisible to others.

4.3.2.10 Global Variables - Super globals

Global variable means that they are always accessible, regardless of scope - and you can access them from any function, class or file.

Examples of PHP global variables are \$GLOBALS, \$_SERVER, \$_REQUEST, \$_POST, \$_GET, \$_FILES, \$_ENV, \$_COOKIE, \$_SESSION etc.

\$GLOBALS is a PHP super global variable which is used to access global variables from anywhere in the PHP script (also from within functions or methods).

PHP stores all global variables in an array called \$GLOBALS[index]. The index holds the name of the variable.

4.3.2.11 Database Functions in PHP

Function	Description
Mysql_connect()	Opens a MYSQL connection
Mysql_select_db()	Sets the active MYSQL database
Mysql_query()	Executes a query on a MYSQL database
Mysql_fetch_object()	Returns a row from a recordset as an object
Mysql_fetch_array()	Returns a row from a recordset as an associate array or/and a numeric array.
Mysql_num_rows()	Returns the number of rows in a recordset.
Mysql_close()	Close the MYSQL connection.

4.3.2.12 PHP Mysql Connection

The syntax for the mysql_connect statement is shown below. This statement will open a connection to the server.

CONNECTIVITY. PHP

```
<?php
$localhost="localhost";
$dbuser="root";
$dbpass="";
$dbname="student"; //the name of the database
```

```

    $connect=mysql_connect($localhost,$dbuser,$dbpass)or
    die("Connection Failure to Database");
    echo "Connected to database server<br>";
    mysql_close($connect);?>

```

CREATE A DATABASE

The CREATE DATABASE statement is used to create a database in MySQL. To get PHP to execute the statement above we must use the mysql_query() function. This function is used to send a query or command to a MySQL connection. A database must be selected before a table can be created. The database is selected with the mysql_select_db() function

The following example shows how you can create a table named "Student", with three columns. The column names will be "FirstName", "LastName" and "Age":

PHP CODE TO CREATE DATABASE AND TABLE INSIDE THAT DATABASE

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{die('Could not connect: ' . mysql_error()); }
// Create database vhse
if(mysql_query("CREATE DATABASE vhse",$con))
{echo "Database created"; }
else
{echo "Error creating database: " . mysql_error(); }
// Create table Student in vhse database
mysql_select_db("vhse", $con);
$sql = "CREATE TABLE Student
(FirstNamevarchar(15),
    LastNamevarchar(15),
    Age int)";
mysql_query($sql,$con);
mysql_close($con);?>

```

INSERT DATA INTO A DATABASE TABLE

The INSERT INTO statement is used to add new records to a database table.

```

<?php
$con = mysql_connect("localhost","root","");

```

```

if (!$con)
{die('Could not connect: ' . mysql_error()); }
mysql_select_db("vhse", $con);
mysql_query("INSERT INTO Student (FirstName, LastName, Age)
VALUES ('Hari', 'Krishnan', '17')");
mysql_query("INSERT INTO staff (FirstName, LastName, Age)
VALUES ('Nikhil', 'Mathew', '17')");
mysql_close($con);?>

```

INSERT DATA FROM A FORM INTO A DATABASE

Now we will create an HTML form that can be used to add new records to the "Student" table.

INSERT.HTML

```

<html>
<body>
<form action="insert.php" method="post">
Firstname: <input type="text" name="firstname" ><br>
Lastname: <input type="text" name="lastname" ><br>
Age: <input type="text" name="age" ><br>
<input type="submit" ></form></body></html>

```

INSERT.PHP

```

<?php
$con = mysql_connect("localhost", "root", "");
if (!$con)
{die('Could not connect: ' . mysql_error()); }
mysql_select_db("vhse", $con);
$sql="INSERT INTO Student (FirstName, LastName, Age) VALUES
($_POST[firstname], $_POST[lastname], $_POST[age])";
if (!mysql_query($sql, $con))
{die('Error: ' . mysql_error()); }
echo "1 record added";
mysql_close($con);?>

```

Select Data From a Database Table

The SELECT statement is used to select data from a database. The following example selects all the data stored in the "Student" table.

SELECT.PHP

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
mysql_select_db("vhse", $con);
$result = mysql_query("SELECT * FROM Student ");
while($row = mysql_fetch_array($result))
{ echo $row['FirstName'] . " " . $row['LastName'];
echo "<br />"; }
mysql_close($con);?>

```

Display The Result in Table Format

The following example selects the same data as the example above, but will display the data in an HTML table

TABLE.PHP

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
mysql_select_db("vhse", $con);
$result = mysql_query("SELECT * FROM Student");
echo "<table border='1'>
<tr>
<th>Firstname</th>
<th>Lastname</th></tr>";
while($row = mysql_fetch_array($result))
{ echo "<tr>";
echo "<td>" . $row['FirstName'] . "</td>";
echo "<td>" . $row['LastName'] . "</td>";
echo "</tr>"; }
echo "</table>";
mysql_close($con);?>

```

The Where Clause

To select only data that matches a specific criteria, add a WHERE clause to the SELECT statement. To get PHP to execute the statement above we must use the mysql_query() function.

The following example will select all rows from the "Student" table, where FirstName='Hari';

SELECTWHERE.PHP

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
mysql_select_db("vhse", $con);
$result = mysql_query("SELECT * FROM Student WHERE FirstName='Nikhil'");
while($row = mysql_fetch_array($result))
{ echo $row['FirstName'] . " " . $row['LastName'];
echo "<br />"; } ?>
```

Update Data in a Database

The UPDATE statement is used to modify data in a database table. The following example updates some data in the "Student" table:

UPDATE.PHP

```
<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
mysql_select_db("vhse", $con);
mysql_query("UPDATE Student SET Age = '17'
WHERE FirstName = 'Hari' AND LastName = 'Krishnan'");
mysql_close($con);?>
```

Delete Data in a Database

The DELETE FROM statement is used to delete records from a database table. The following example deletes all the records in the "Student" table where LastName='Krishnan':

DELETE.PHP

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
mysql_select_db("vhse", $con);
mysql_query("DELETE FROM Student WHERE LastName='Krishnan'");
mysql_close($con);
?>

```

Now we are learnt how to manipulate data in database using PHP program. Here is an example to insert data in ContactUs table using PHP program. HTML form used for this purpose is the file created in 4.3.2.6 example.

For that change the attribute of form as follows <form method=post action=contactus.php>

Program 3.23

PHP program for creating database and table

```

<?php
$con = mysql_connect("localhost","root","");
if (!$con)
{ die('Could not connect: ' . mysql_error()); }
// Create database animal
if (mysql_query("CREATE DATABASE animal",$con))
{ echo "Database created"; }
else
{ echo "Error creating database: " . mysql_error(); }
// Create table Student in vhse database
mysql_select_db("animal", $con);
$sql = "CREATE TABLE contactUs (
FirstName varchar(15), LastName varchar(15),emailId varchar(30),Sex
varchar(10),
Query varchar(50),Heardsite varchar(30))";
mysql_query($sql,$con); mysql_close($con); ?>

```

Program 3.24

PHP program for creating database and table

```

<?php
$a=$_POST["Firstname"];

```

```

$b=$_POST["Lastname"];
$c=$_POST["emailid"];
$d=$_POST["sex"];
$e=$_POST["query"];
$con = mysql_connect("localhost","root","");
if(!$con)
{
die('Could not connect: ' . mysql_error());
}
mysql_select_db("animal", $con);
mysql_query("INSERT INTO Contactus VALUES ('$a', '$b', '$c', '$d', '$e')");
mysql_close($con);
?>

```

ASSESSMENT ACTIVITY

- Create database students.
- Create a table passouts which contains fields in alumni.html.
- Insert, update and delete records of passouts using alumni.html.

EXTENDED ACTIVITY

By using the php and database connectivity, update example 4.2.7.3 as follows. Place 2 button SignUp and SignIn in the form. SignUp is used for login as new user and SignIn is used for existing users. Use appropriate code for database connectivity and updation.

List of Practicals

1. Design a login page using javascript
2. Find perimeter of a circle using javascript
3. Write a query to display the Total Mark, Average Mark, Maximum Mark, Minimum Mark
4. Write a query to display the number of students in the Course "CSIT"
5. Write a JavaScript program to find the sum of two numbers
6. Create a webpage that checks the eligibility for voting
7. Create a webpage to print the day of a week using switch
8. Create a webpage that display the first 10 odd numbers
9. Create a webpage that display the multiplication table of a number

10. Create an application form of birth certificate
11. Write a PHP program to display a message
12. Design a web page that redirects to another page after an interval of time
13. Design a page with forward and backward buttons for loading other pages
14. Design an application form with submit button and display the contents entered by user in another page while clicking submit button
15. Design a bill format and do the appropriate calculations
16. Create a database in MySQL and connect to a PHP page which shows a successful message on successful connection
17. Design a login page and home page for a website using database
18. Design a web page that implements insert, delete and update functions in a table
19. Design a web page that connects more than two tables and display contents
20. Develop a PHP project for online book store
21. Develop a PHP project for user information retrieval system using unique ID [Registration, updating, deletion and retrieval of citizens' information]
22. Develop a PHP project for payroll management of a company
23. Practice on domain name registration
24. Create a simple webpage and that makes use of free hosting
25. Design and host the website of your school

TE Questions

Type: Objective

1. Odd one out. Justify reason.
a. VB Script b. JavaScript c. SQL d. PHP
2. Keyword used to define a constant is
3. State whether true or false. "PHP is case sensitive".
4. In PHP multiline comment can enclosed in.....
5. Select invalid variable name with reason.
a. \$123 b. \$FirstName c. \$Last Name d. Number1

Type: Short Answer

1. Predict the output of the following expression in php.
a. $\$a=2+6/2*4-5$;
Echo \$a;
b. $\$a="Computer"$;

```

$b="Science";
Echo $a.$b;
c. $c=10;
$c+=5;
Echo $c

```

2. Predict the output of the following.

```

$a=10;
if($a%2==0) echo $a."is even";
else echo $a. "is odd";

```

3. Replace the code with if..elseif..else.

```

<?php
$num=$_POST[num];
switch($num){
case 1:
echo "One";
break;
case 2:
echo "Two";
break;
default:
echo "Enter 1 or 2";}
?>

```

3. Predict the output of the following functions.

```

"Echo strtoupper("computer");
"Echo strev("science");
"Echo strlen("VHSE");

```

4. Compare Associative array and indexed array.

Type: Essay

- Write PHP code to check whether the given number is prime or not using do..while.
- Write PHP program to find factorial of a number using function without argument and return value
- Write a PHP program to find the value of nCr using functions.
[$nCr = n!/(r!(n - r)!)$]

Unit - 4

Web Hosting

Now we studied in detail how to create static and dynamic web pages. In this unit they are introduced to the concepts of webhosting. Starting with the different types of webhosting, learners are guided through the process of hosting a site. How to buy a web hoisting space, how to register a domain and what are the importance of FTP client software are etc. are also discussed. This chapter should provide an encouragement for every student to try-out something with web hosting. It should give a strong foundation for the learner to move to more sophisticated web tools. Necessary demonstration and hands-on session may be given to learners.

Learning Outcomes

The learner:

- Compare different types of web hoisting
- Explain the steps of buying hoisting space
- Describe the concepts of domain name registration
- Explain the use of FTP software

4.4.1 Types of Hosting

Web hosting: It is a Service allows individuals and organizations to make their website accessible to the public. Organizations that provide resources such as disk space, bandwidth, etc. are referred to as "Web Hosts" or "Web Hosting Providers". Webhosts usually keep their infrastructure in a placed commonly referred to as a "DataCenter".

1. Free hosting

Free web hosting is best suited for small sites with low traffic, like personal sites. In a free hosting environment, connection speed is slow, website can be down frequently, and advertising banners is automatically added to your website. Also technical support is limited and technical options are few. Some companies insist to purchase domain name to receive free hosting services from them, while others offer you a free subdomain under them, such as [ourname.webhost.com]

2. Shared (virtual) hosting

Shared hosting is very cost effective. In shared hosting, web site gets its own domain name, and is hosted on a powerful server along with maybe 100 other web sites. In

a shared hosting environment, website owners shared one server. This includes sharing the physical server and the software applications within the server. Shared hosting services are affordable because the cost to operate the server is shared between you and these other owners. There are, however, a number of down sides, such as being slower. Reduced security due to many sites on one server and Restricted database support.

3. Dedicated hosting

With dedicated hosting, web site is hosted on a dedicated server. Dedicated hosting is the most expensive option. This option is best suited for large web sites with high traffic, and web sites that use special software. This allows for faster performance, and have all the server's resources entirely, without sharing with other website owners. However, this also means that we will be responsible for the cost of server operation. Dedicated hosting is very powerful and secure, with almost unlimited software solutions.

4. Collocated hosting

Collocation means "co-location". Collocated hosting lets you place your own web server on the premises (locations) of a service provider. You will be responsible for the server itself. An advantage of this type of hosting service is that have full control of the web server. This is pretty much the same as running your own server in your own office, only that it is located at a place better designed for it. Most likely an ISP will have dedicated resources like high-security against fire and vandalism, regulated backup power, dedicated Internet connections and more.

4.4.2 Buying Hosting Space

Web hosting is provided by Internet service providers as a general Internet access plan. There are free and paid web service providers offering web hosting . A customer can decided which type of space is needed for hosting according to their requirements. Considerations include database server software, scripting software, and operating system. Most hosting providers provide Linux-based web hosting which offers a wide range of different software. A typical configuration for a Linux server is the LAMP platform: Linux, Apache, MySQL, and PHP/Perl/Python.

4.4.3 Domain Name Registration

The first step is to select a service provider as per your criteria. Then decide a domain name and search the desired domain name is available or not. If it is not registered previously, it can be considered for registration. The next step is to register

that domain. For registration you have to give your name and contact information, terms for how long you will like to have the domain for yourself. There will be a registration fee.

Main page of who.is, ip address and domain name search website.

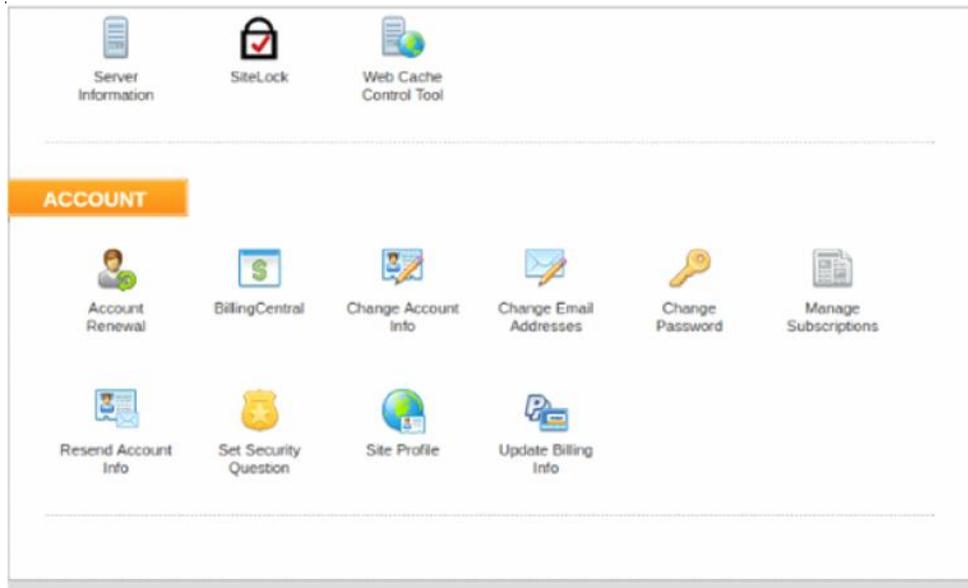
The screenshot shows the iPage website header with the logo and the text "WEB HOSTING LEADER" and "Our platform serves over 1,000,000 customers". Below the header, the main heading is "Step 1 : Select Domain". Underneath, it says "Step 1: Select Domain > Step 2: Enter Your Billing Information". The main content area is titled "Select a Domain" and contains two radio button options: "Register a new domain." and "I plan to use a domain I already own." Below these options is a text input field for "Domain Name:" with the value "www.mynewwebsite.com" and a "Check Availability" button.

Create a custom domain name for the website on ipage.com.

The screenshot shows the "Purchase Information" section. It lists the following details: "iPage Essential Hosting Plan: \$1.99 per month (billed \$47.76 for 2 years)", "Cancellation Policy: 30-Day Money-Back Guarantee!", "Marketing Credits: Over \$450 in Freebies", "Instant Activation: INCLUDED FREE!", "Setup Fee: INCLUDED FREE!", "Domain Registration: INCLUDED FREE!", and "Domain Privacy: \$9.99/year — Protect your identity. See how...".

Billing section and account creation in ipage.com

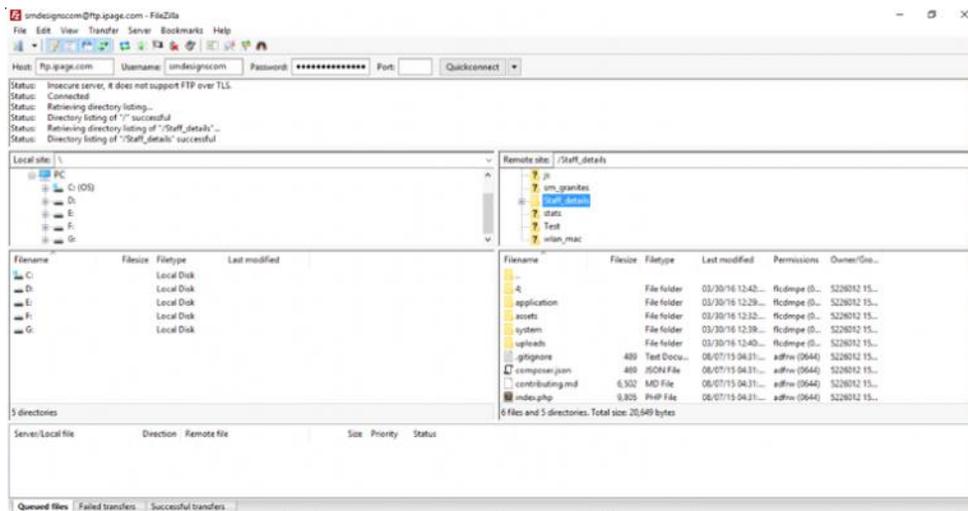
The screenshot shows the iPage website header with the logo and the text "WEB HOSTING LEADER" and "Our platform serves over 1,000,000 customers". Below the header, the main heading is "Step 2 : Enter Your Billing Information". Underneath, it says "Step 1: Select Domain > Step 2: Enter Your Billing Information". The main content area is titled "Step 2 : Enter Your Billing Information" and contains a message: "mynewwebsite.website will be live on iPage in just a few minutes ...". Below this message, there are two sections: "Contact Information" and "Payment Information". The "Contact Information" section has input fields for "First Name:", "Last Name:", "Email Address:", "Phone Number:", "Street Address:", "City:", "State/Province:", "Zip Code:", and "Country:". The "Payment Information" section has a "Payment Type:" section with radio buttons for "Credit Card" and "PayPal", and input fields for "Card Number:", "Expiration Date:", and "Card Security Code:".



Control panel of ipage webhosting

4.4.4 FTP client Software

After buying a space on a hosting server and a domain name, we need to transfer the file of the website from our computer to the web server. This requires an FTP client software.



Extended Activity

- Create a website using HTML, JAVASCRIPT AND PHP and also Host the website
- Develop a website called animal.com

Create a "login.html" page with the help of JAVASCRIPT

By using the php and database connectivity, manipulate the animal details created in the above unit (add animal description, update details, etc..)

Above created animal.html page can be hosted using following steps

Step 1: Buying a host page from any of the web server

Step 2: Register a domain name (try to get the domain name "animal.com")

Step 3: After login to the webserver we can link the domain name with our web space

Step 4: Transfer the webpages from our computer (local host) to the webserver using FTP software.

Step 5: Access the website from any of the browser ("www.animal.com")

- Develop a new website for your school and host it

TE Questions**Type: Objective**

1. Shared hosting is also known as hosting
2. Name the hosting technique which allow us to place our webserver in our own location.
3. Most expensive hosting technique is
4. is used to transfer the files of the website from our computer to the webserver.

Type: Short Answer

1. Write a short note on domain name registration.
2. Explain the importance of FTP client software.

Type: Essay

1. Explain different types of hosting.

Unit - 5

TRENDS IN COMPUTING TECHNOLOGY

We live in a time when discoveries in science and technology are happening at a pace that makes it difficult to stay informed. The full potential of these changes are continuously evolving especially in computing technology. IEEE Computer Society journals, magazines, and conferences are continually at the forefront of current technology trends. Below is a list of technology which will be focusing on coming years:

Learning Outcomes

The learner:

- Understand the concept of mobile network
- Analyse different generation of networks
- Distinguish different wireless networking technologies
- Identify mobile application development
- Identify the importance of different computing technologies
- Explain IoT
- Explain the applications of Artificial Intelligence

4.5.1 Mobile Technologies

- Mobile technology is the technology used for cellular communication.

Advantages of mobile technology

- Mobile technology is portable used in laptop, tablets, net book computers, smart phones
- Mobile technology is used in Global Positioning System (GPS) devices and wireless debit/credit card payment terminals
- Mobile devices can be enabled to use a variety of communications technologies,
- It is making possible to network the mobile device to a home office or the internet while travelling.

Disadvantages of mobile technologies

- Mobile technology is expensive in setting up the equipment and training.
- Mobile IT devices may suffer from unauthorised access of valuable data to people if the proper precautions are not taken.

4.5.2 Generation of Networks

The different generation of cellular communications are given below

2G (Second Generation)

2G were launched on the GSM standard in Finland by Radiolinja in 1991 (1G systems were introduced in 1981/1982.). The primary benefits of 2G wireless telephone networks over their predecessors were the phone conversations were digitally encrypted. 2G introduced data services for mobile, starting with SMS text messages and expand to provide the services such as text messages, picture messages and MMS (Multi Media Messages).

3G (Third Generation)

The first 3G networks were introduced in 1998. 3G finds application in wireless voice telephony, mobile Internet access, fixed wireless Internet access, video calls and mobile TV. It provides information transfer rate of at least 200 kbit/s. The International Telecommunications Union (ITU) defined the third generation (3G) of mobile telephony standards IMT-2000 to facilitate growth, increase bandwidth, and support more diverse applications.

4G (Fourth Generation)

Fourth generation (4G) networks were introduced in 2008. 4G system must provide capabilities defined by ITU (International Telecommunications Union) in IMT (International Mobile Telecommunications). Potential and current applications include amended mobile web access, IP telephony, gaming services, high-definition, mobile TV, video conferencing, 3D television, and cloud computing. The speed requirements for 4G service at 100 megabits per second (Mbit/s) for high mobility communication

4.5.3 Wireless Networking Technologies

CDMA AND GSM

GSM (Global System for Mobile communications) and CDMA (Code Division Multiple Access) are competing wireless technologies with GSM enjoying about an 82% market share globally. CDMA refers to digital cellular telephony systems that use this multiple access scheme. CDMA allows multiple communication at the same time while GSM splits data into cells and transmit with different frequencies.

Data Transfer Methods in GSM VS. CDMA

Main difference between GSM and CDMA is in the data transfer methods. GSM's high-speed wireless data technology, GPRS (General Packet Radio Service), usually

offers a slower data bandwidth for wireless data connection than CDMA's high-speed technology, which has the capability of providing ISDN (Integrated Services Digital Network)-like speeds of as much as 144Kbps (kilobits per second).

Comparison chart of CDMA AND GSM

	CDMA	GSM
Stands for	Code Division Multiple Access	Global System for Mobile communication
Storage Type	Internal Memory	SIM (subscriber identity module) Card
Global market share	25%	75%
Dominance	Dominant standard in the U.S.	Dominant standard worldwide except the U.S.
Network	There is one physical channel and a special code for every device in the coverage network. Using this code, the signal of the device is multiplexed, and the same physical channel is used to send the signal.	Every cell has a corresponding network tower, which serves the mobile phones in that cellular area.
International roaming	Less Accessible	Most Accessible
Frequency band	Single (850 MHz)	Multiple (850/ 900/ 1800/ 1900 MHz)
Network service	Handset specific	SIM specific. User has option to select handset of his choice.

Subscriber Identity Module (SIM Card)

The Subscriber Identity Module (SIM) is a small smart card which contains both programming and information. The Subscriber Identity Module (SIM) can be used to store user-defined information such as phonebook entries. One of the advantages of the GSM architecture is that the SIM may be moved from one Mobile Station to another. This makes upgrades very simple for the GSM telephone user.

International Mobile Equipment Identity (IMEI)

The IMEI (International Mobile Equipment Identity) is a unique 17 or 15 digit code used to identify an individual mobile station to a GSM or UMTS network. The IMEI number provides an important function; it uniquely identifies a specific mobile

phone being used on a mobile network. The IMEI is a useful tool to prevent a stolen handset from accessing a network and being used to place calls. Mobile phone owners who have their phones stolen can contact their mobile network provider and ask them to disable a phone using its IMEI number. With an IMEI number, the phone can be blocked from the network quickly and easily.

Wireless Lan Technologies

Wi-Fi

Wi-Fi or WiFi is a technology that allows electronic devices to connect to a wireless LAN (WLAN) network, mainly operate in the 2.4GHz frequency band and have a typical range of around 500 feet (with clear line of sight), whereas indoors a user can expect around 150 feet. Access to a WLAN can be protected by password.

Devices which can use Wi-Fi technology include personal computers, video-game consoles, smartphones, digital cameras, tablet computers and digital audio players. Wi-Fi compatible devices can connect to the Internet via a WLAN network and a wireless access point. Such an access point (or hotspot) has a range of about 20 meters (66 feet) indoors and a greater range outdoors. Hotspot coverage can be as small as a single room with walls that block radio waves, or as large as many square kilometres achieved by using multiple overlapping access points.

WiMAX (Worldwide Interoperability for Microwave Access)

WiMAX is designated as the Metropolitan Area Network (MAN) technology that can connect IEEE 802.11 (Wi-Fi) hotspots with each other and to other parts of the Internet. However, the field of uses is broader and overlaps those for mobile WAN (wide area networks) and WLANs.

RFID (Radio-frequency Identification)

Radio-frequency identification (RFID) is the use of an object applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several metres away and beyond the line of sight of the reader.

Most RFID tags contain at least two parts. One is an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency (RF) signal, and other specialized functions. The second is an antenna for receiving and transmitting the signal.

4.5.4 Mobile APP Development

Mobile application development is a process by which application software is developed for handheld devices, such as personal digital assistants, enterprise digital assistants or mobile phones. These applications can be pre-installed on phones during manufacturing platforms, or delivered as web applications. Mobile app development has been steadily growing, both in terms of revenues and jobs created.

Creating Mobile Applications

The first step in creating mobile applications is a basic understanding of your options. Mobile applications come in two formats: Native applications and mobile web applications.

Native applications

A native mobile application is simply a piece of software for smartphones and tablets. Native applications are built specifically for each mobile platform and installed on the device itself. Just like PC software doesn't work on a Mac, each native mobile app only works on the platform for which it was built. If you want native apps to work across all mobile platforms, you must build separate versions for each platform.

Web applications

A mobile web application is a web application formatted for use on a smartphone or tablet and accessed through the device's web browser. Since mobile web applications are accessed through the browser without requiring installation on each device, they are platform independent.

The biggest difference between the two options: Native applications are installed directly on each device while web applications are served from a central location and accessed through a web browser. Both options come with their own unique drawbacks and benefits.

Questions to ask before creating mobile apps

1. How many platforms do you need to support?
2. Do you need to use hardware sensors?
3. How important is security?
4. What's the purpose of your app?
5. How important is data integration?

Mobile app is the future of business. Smartphone and tablet sales are on the rise and businesses are finally jumping on board. However, choosing the right path is a challenging task for business just stepping out into mobile territory.

Five important factors that will impact your mobile application decision:

- If you want apps that work across multiple platforms, mobile web apps are a better option.
- If you want apps that access the device's camera or microphone, native apps are a better option.
- If security is important, mobile web apps are a better option.
- If you want to sell your apps, native apps are a better option.
- If you want apps that integrate with existing systems and databases, mobile web apps are a better option.

As part of the development process, Mobile User Interface (MUI) Design is also an essential in the creation of mobile apps. Mobile UI considers constraints & contexts, screen, input and mobility as outlines for design. The user is often the focus of interaction with their device, and the interface entails components of both hardware and software. User input allows for the users to manipulate a system, and device's output allows the system to indicate the effects of the users' manipulation.

Mobile UIs, or front-ends, rely on mobile back-ends to support access to enterprise systems. The mobile back-end facilitates data routing, security, authentication, authorization, working off-line, and service orchestration.

Mobile Operating System

A mobile operating system (or mobile OS) is an operating system for smart phones, tablets, PDAs, or other mobile devices. Mobile operating systems combine features of a personal computer operating system with other features useful for mobile or handheld use. The most popular OS's for mobile devices are Apple's iOS and Google's Android. Other Operating systems commonly used are RIM's BlackBerry OS and Microsoft's Windows Phone. Today mobile devices with a proper OS are called smart phones and users have a wide choice of applications, such as games, productivity apps, communication or social media apps, digital maps, etc.

4.5.5 Computing Techniques

Cloud Computing

Cloud computing, also on-demand computing, is a kind of Internet-based computing that provides shared processing resources and data to computers and other devices on demand. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centers.

Cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand.

Cloud computing has become a highly demanded service or utility due to the advantages of high computing power, cheap cost of services, high performance, scalability, accessibility as well as availability. The goal of cloud computing is to allow users to take benefit from all of these technologies, without the need for deep knowledge about or expertise with each one of them.

The main enabling technology for cloud computing is virtualization. Virtualization provides the ability required to speed up IT operations, and reduces cost by increasing infrastructure utilization.

Quantum Computing

Quantum computing studies theoretical computation systems (quantum computers) that make direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data. Quantum computers are different from digital electronic computers based on transistors. Whereas digital computers require data to be encoded into binary digits (bits), quantum computation uses quantum bits (qubits), which can be in superposition of states. A quantum Turing machine is a theoretical model of such a computer, and is also known as the universal quantum computer. Quantum computers share theoretical similarities with non-deterministic and probabilistic computers. Large-scale quantum computers would be able to solve certain problems much more quickly than any classical computers.

A single qubit can represent a one, a zero, or any quantum superposition of those two qubit states; a pair of qubits can be in any quantum superposition of 4 states, and three qubits in any superposition of 8 states. In general, a quantum computer with qubits can be in an arbitrary superposition of up to different states simultaneously (this compares to a normal computer that can only be in one of these states at any one time).

Nano Computing

Nano computing describes computing that uses extremely small, or nanoscale, devices. The integrated circuits (IC) industry, however, looks to the future to determine the smallest electronic devices possible within the limits of computing technology. Nano computing technology has the potential for revolutionizing the way that computers are used. In order to achieve this goal, major progress in device technology, computer architectures, and IC processing must first be accomplished.

Nanocomputing research involves the study of very small electronic devices and molecules, their fabrication, and architectures that can benefit from their inherent electrical properties.

4.5.6 Internet on Small Things (Iot)

The Internet of Things (IoT) is the network of physical objects-devices, vehicles, buildings and other items-embedded with electronics, software, sensors, and network connectivity that enables these objects to collect and exchange data. The IoT allows objects to be sensed and controlled remotely across existing network infrastructure, creating opportunities for more direct integration of the physical world into computer-based systems, and resulting in improved efficiency, accuracy and economic benefit. Each thing is uniquely identifiable through its embedded computing system but is able to interoperate within the existing Internet infrastructure. Experts estimate that the IoT will consist of almost 50 billion objects by 2020.

IoT is expected to offer advanced connectivity of devices, systems, and services that goes beyond machine-to-machine (M2M) communications and covers a variety of protocols, domains, and applications.

"Things," in the IoT sense, can refer to a wide variety of devices such as heart monitoring implants, biochip transponders on farm animals, electric clams in coastal waters, automobiles with built-in sensors, DNA analysis devices for environmental/food/pathogen monitoring or field operation devices that assist fire fighters in search and rescue operations. Based on the application domain, IoT products can be classified broadly into five different categories: smart wearable, smart home, smart city, smart environment, and smart enterprise.

Sensing Network

Wireless sensor networks (WSN) use autonomoussensors to monitor physical or environmental conditions, such as temperature, sound, pressure, etc. and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling control of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

BIG DATA ANALYSIS

Big Data is a term that is used to describe data that is high volume, high velocity, and/or high variety; requires new technologies and techniques to capture, store, and analyze it; and is used to enhance decision making, provide insight and discovery, and support and optimize processes.

Big data analytics is the process of examining large data sets containing a variety of data types -- i.e., big data -- to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information.

Examples of big data analytics: A company introducing a New Coffee Product.

Company was introducing a new coffee product but was concerned that customers would find its taste too strong. The morning that the coffee was rolled out, Company monitored blogs, Twitter, and niche coffee forum discussion groups to assess customers' reactions. Mid-morning Company discovered that although people liked the taste of the coffee, they thought that it was too expensive. Company lowered the price, and by the end of the day all of the negative comments had disappeared. Compare this fast response with a more traditional approach of waiting for the sales reports to come in and noticing that sales are disappointing. A next step might be to run a focus group to discover why. Perhaps in several weeks Company would have discovered the reason and responded by lowering the price.

4.5.7 Artificial Intelligence

Artificial intelligence (AI) is the intelligence exhibited by machines or software. Ultimate AI would be a recreation of the human thought process -- a man-made machine with our intellectual abilities. This would include the ability to learn just about anything, the ability to reason, the ability to use language and the ability to formulate original ideas.

The real challenge of AI is to understand how natural intelligence works. Developing AI isn't like building an artificial heart -- scientists don't have a simple, concrete model to work from. We do know that the brain contains billions and billions of neurons, and that we think and learn by establishing electrical connections between different neurons. But we don't know exactly how all of these connections add up to higher reasoning, or even low-level operations. The complex circuitry seems incomprehensible.

Applications Of AI

Artificial intelligence has been used in a wide range of fields including medical diagnosis, stock trading, robot control, law, remote sensing, scientific discovery and toys. Artificial neural networks are used as clinical decision support systems for medical diagnosis, such as in Concept Processing technology in EMR software. Another application in medical science is the detection of a tumour.

Knowledge Engineering

A field within artificial intelligence that develops knowledge-based systems. Such systems are computer programs that contain large amounts of knowledge, rules and reasoning mechanisms to provide solutions to real-world problems. A major form of knowledge-based system is an expert system, one designed to emulate the reasoning processes of an expert practitioner (i.e. one having performed in a professional role for very many years). One of the first examples of an expert system was MYCIN, an application to perform medical diagnosis. In the MYCIN example, the domain experts were medical doctors and the knowledge represented was their expertise in diagnosis.

Natural Language Processing

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. Some of the most commonly researched tasks in NLP are Automatic summarization, Machine translation, Natural language generation, Question answering, Speech recognition, Speech processing etc. Other tasks include Native Language Identification, Text simplification, Text-to-speech, Text-proofing, Query expansion etc.

ROBOTICS

Artificial Intelligence is the most exiting field in Robotics. Robotics deal with automated machines that can take the place of humans in dangerous environment or manufacturing process. It resembles humans in appearance and behaviour and uses the Artificial Intelligent techniques.

Assessment Activities

Prepare power point presentation about Cloud computing, Quantum computing and nano computing

TE Questions**I. Objective Type Questions**

1. Expand GPS
2. ----- Technology in mobile communication enjoying high market share globally
(A). GSM (B). CDMA (C). CLOUD (D). All of these
3. Example for mobile operating system is
(A). iOS (B). Android (C). BlackBerry (D). All of these
4. -----describes computing that uses extremely small devices.
5. ----- is an examples of an expert system used to perform medical diagnosis.

ii. Short Answer

1. Compare 2G and 3G in mobile technology?
2. Write a short note on SIM
3. Explain the advantages of Cloud computing
4. Discuss the main application areas of IoT?
5. Discuss any 4 application areas of AI?

III. Essay

1. Compare GSM and CDMA?
2. Explain any 3 wireless communication technologies

List of Reference

- Chew, Frederick F: *The Java/C++ Cross-Reference Handbook*, Prentice-Hall, 1998.
- Deitel HM and Deitel PJ: *C++ How to Program, Third Edition*, Prentice Hall, 2001.
- Graham, Neill: *Thinking in C++*, McGraw-Hill, 1991
- Josuttis, Nicolai: *Object-Oriented Programming in C++*, John Wiley & Sons, 2003.
- Kernighan B and Ritchie D: *The C Programming Language*, Prentice Hall,
- Lischner R: *C++ in a Nutshell*, O'Reilly, 2003. Meyers, Scott: *Effective*
- Skiena, Steven S.: *The Algorithm Design Manual*, Springer Telos, 1998.
- *Engineering A Hands-On Approach*, Prentice Hall, 2004.
- *An Introduction to DBMS Bipin C Desai*, West Publishing Company
- *Concept of Database Management Systems Dbms 2nd Edition*, Pallaw
- *Database Management System*, Tata McGraw-Hill Education, 2011
- *HTML, XHTML, and CSS (Visual Quickstart Guide)*, 2002
- *HTML 5 and CSS 3 Made Simple by I Bayross* Bpb Publications
- *Learning Php, Mysql & Javascript Paperback*, Shroff Publishers & Distributors Pvt. Ltd.,
- www.w3schools.com
- www.wikipedia.org