COMPUTER APPLICATION (COMMERCE)

PREPARED BY:

VIJU K B
SSHSS MOORKANAD

MOHAMMED JASIM K
DHOHSS POOKKARATHARA

SREEKANTH K
GJHSS NADUVATTOM

SHIJU P B
GHSS KUTTIPPURAM

ABDUL MAJEED C
GHSS PATTIKKAD

RADHIKA C V
GHSS MATTUMMAL

Chapter 1
Review of C++ Programming

- **Tokens:** ഒന്നോ അതിലധികമോ അക്ഷരങ്ങൾ ചേർന്ന് നിർമിക്കപ്പെട്ടിട്ടുള്ള പ്രോഗ്രാമിന്റെ ഏറ്റവും ചെറിയ ഘടകമാണ് **ടോക്കൺ.**
- Keywords, Identifiers, Literals, Punctuators, Operators എന്നിവയാണ് സി++ ലെ ടോക്കണുകൾ.

1. **Keywords:** കംപൈലർ പ്രത്യേക അർത്ഥം കൽപിച്ചിരിക്കുന്ന പദങ്ങളാണ് **കീവേഡുകൾ.**
2. **Identifiers:** മെമ്മറി ലൊക്കേഷനുകൾ, ഫംഗ്ഷനുകൾ മുതലായവയ്ക്ക് പേരിടാൻ user നിർവചിച്ച പദങ്ങളാണ് **ഐഡന്റിഫയറുകൾ.**
3. **Literals/Constants:** പ്രോഗ്രാം എക്സിക്യൂഷൻ സമയത്ത് മൂല്യം ഒരിക്കലും മാറാത്ത ടോക്കണുകളാണ് **ലിറ്ററലുകൾ / കോൺസ്റ്റന്റ്സ്.**

   ലിറ്ററലുകളെ നാലായി തരം തിരിച്ചിരിക്കുന്നു:

   (i) integer constants: അക്കങ്ങളാൽ മാത്രം രൂപീകരിച്ച ടോക്കണുകൾ. Eg:- 101

   (ii) floating point constants: ഭിന്ന സംഖ്യകൾ. Eg:- 10.5

   (iii) character constants: ഒരു ജോഡി Single quotes നുള്ളിൽ ഉൾപ്പെടുത്തിയിരിക്കുന്ന ഒരൊറ്റ ക്യാരക്ടറിനെ **ക്യാരക്ടർ കോൺസ്റ്റന്റ്** എന്ന് വിളിക്കുന്നു. Eg:- 's'

   (iv) string constants: ഒരു ജോഡി double quotes നുള്ളിൽ ഉൾപ്പെടുത്തിയിരിക്കുന്ന ഒന്നോ അതിലധികമോ ക്യാരക്ടറുകളുടെ ശ്രേണിയെ **സ്ട്രിംഗ് കോൺസ്റ്റന്റ്** എന്ന് വിളിക്കുന്നു. Eg:- "Hello"
4. **Punctuators:** കോമ (,), സെമി കോളൻ (;), ഹാഷ് (#) മുതലായ പ്രത്യേക ചിഹ്നങ്ങൾ.
5. **Operators:** കംപൈലറോട് ഒരു പ്രത്യേക പ്രവർത്തിയെക്കുറിച്ച് പറയുന്നതിന് ഉപയോഗിക്കുന്ന ചിഹ്നങ്ങളാണ് **ഓപ്പറേറ്ററുകൾ.**

   Eg:- Arithmetic operators (+, -, *, /, %), Relational operators (<, <=, >, >=, ==, !=) and logical operators (&&, ||, !).

- **Data types:** ഒരു വേരിയബിളിന് സംഭരിക്കാനാകുന്ന ഡാറ്റയുടെ വലുപ്പത്തെയും തരത്തെയും നിർവചിക്കുന്നു.
- ഡേറ്റ ടൈപ്പുകളെ **Fundamental data types**, **User-defined data types** എന്നിങ്ങനെ രണ്ടായി തരം തിരിച്ചിരിക്കുന്നു.
- സി++ ലെ fundamental data type കളാണ് **char, int, float, double, void** എന്നിവ.

Control Statements

- പ്രോഗ്രാം എക്സിക്യൂഷന്റെ സാധാരണ ഒഴുക്ക് മാറ്റുന്നതിന് **കൺട്രോൾ സ്റ്റേറ്റ്മെന്റുകൾ** ഉപയോഗിക്കുന്നു.
- കൺട്രോൾ സ്റ്റേറ്റ്മെന്റുകളെ രണ്ടായി തിരിച്ചിരിക്കുന്നു:

(i) **selection statements:** ചില നിബന്ധനകളെ അടിസ്ഥാനമാക്കി തീരുമാനങ്ങൾ എടുക്കുവാൻ **സെലക്ഷൻ സ്റ്റേറ്റ്മെന്റ്സ്** ഉപയോഗിക്കുന്നു. (if, switch)

(ii) **looping statements:** ഒന്നോ അതിലധികമോ സ്റ്റേറ്റ്മെന്റുകൾ ആവർത്തിക്കുവാനായി **ലൂപിംഗ് സ്റ്റേറ്റ്മെന്റുകൾ** ഉപയോഗിക്കുന്നു. (while loop, for loop, do...while loop.)

- **Elements of a loop:** 1. Initialization 2. Test expression 3. Update statement 4. Body of the loop
- **Entry-controlled loop:** ആദ്യം ടെസ്റ്റ് എക്സ്പ്രഷൻ പരിശോധിക്കുന്നു, അത് ശരിയാണെങ്കിൽ ലൂപ്പിന്റെ ബോഡി എക്സിക്യൂട്ട് ചെയ്യും. Eg: for loop, while loop
- **Exit controlled Loop:** ബോഡി ഒരു തവണ എക്സിക്യൂട്ട് ചെയ്തതിനു ശേഷമാണ് ടെസ്റ്റ് എക്സ്പ്രഷൻ പരിശോധിക്കുന്നത്. Eg: do...while loop

Nested Loops

- ഒരു ലൂപ്പിനുള്ളിൽ മറ്റൊരു ലൂപ്പിനെ സ്ഥാപിച്ചിരിക്കുന്നതിനെ **നെസ്റ്റഡ് ലൂപ്** എന്ന് വിളിക്കുന്നു..

  Eg: a 'for' loop inside a 'while' loop

Chapter 2
Arrays

- **Array:** തുടർച്ചയായ മെമ്മറി സ്ഥാനങ്ങളിൽ ശേഖരിച്ചു വച്ചിട്ടുള്ള ഒരേ ഇനത്തിലുള്ള ഡാറ്റകളുടെ സമൂഹമാണ് **അറെ.**
- ഒരു പേരിൽ ഒരേ ഇനത്തിലുള്ള ഒരു കൂട്ടം വിലകൾ ശേഖരിക്കുന്നതിനായി **അറേകൾ** ഉപയോഗിക്കുന്നു..

Syntax for declaring an array:

data_type array_name[size];

Eg:- int num[10]; // ഈ അറേയ്ക്ക് 10 പൂർണ്ണ സംഖ്യകൾ സംഭരിക്കാൻ കഴിയും.

Array initialisation

- അറേയുടെ ഡിക്ലറേഷൻ സമയത്തു തന്നെ അവയുടെ പ്രാരംഭ വിലകൾ നൽകുവാൻ കഴിയും, അറേയിലെ അംഗങ്ങളെ ബ്രാക്കറ്റിനുള്ളിൽ എഴുതണം.

  Eg:- int score[5] = {8, 7, 2, 9, 5};

  score[0] ലേക്ക് 8, score[1] ലേക്ക് 7, score[2] ലേക്ക് 2, score[3] ലേക്ക് 9, score[4] ലേക്ക് 5 ഉം പ്രാരംഭ വിലകളായി സൂക്ഷിക്കുന്നു.

Accessing elements of arrays

- ഒരു അറേ അംഗത്തെ അറേയുടെ പേരും ഇൻഡെക്സ് നമ്പറും ഉപയോഗിച്ച് ആക്സസ് ചെയ്യാൻ കഴിയും. Eg:- cout << score[2]; // Prints 2
- ഒരു **ഫോർ ലൂപ്** ഉപയോഗിച്ച് ഒരു അറേയിലെ അംഗങ്ങളെ എളുപ്പത്തിൽ ആക്സസ് ചെയ്യാൻ കഴിയും.

  Eg:
  ```
  for(i=0; i<5; i++)
      cout<<score[i];        // Prints  8  7  2  9  5
  ```

String handling using arrays

- ഒരു ജോടി double quotes നുള്ളിലെ കാരക്റ്ററുകളുടെ ഒരു ശ്രേണിയാണ് **String**. ഒരു സ്ട്രിംഗ് സംഭരിക്കുന്നതിന് ഒരു **character array** ഉപയോഗിക്കാം.

  Eg:- char my_name[10]; can store a string of 9 characters plus one **null character**.
- ഒരു **null കാരക്ടർ '\0'** സ്ട്രിംഗിന്റെ അവസാനം കൂട്ടിച്ചേർക്കപ്പെടുന്നു. ഈ കാരക്ടർ ***സ്ട്രിംഗ് ടെർമിനേറ്ററായി*** ഉപയോഗിക്കുന്നു.
- cin >> my_name; എന്ന സ്റ്റേറ്റ്മെന്റ് ഉപയോഗിച്ച് my_name എന്ന വേരിയബിളിൽ ഒരു സ്ട്രിംഗ് സൂക്ഷിക്കാൻ കഴിയും. എന്നാൽ ഈ സ്റ്റേറ്റ്മെന്റ് ഉപയോഗിച്ച് വൈറ്റ് സ്പേസ് അടങ്ങിയ സ്ട്രിങ്ങുകൾ ഇൻപുട്ട് ചെയ്യുവാൻ കഴിയില്ല.
- (i.e, if we input the string "Maya Mohan", only the first word "Maya" is stored.)
- ഇതിനു പരിഹാരമായി **gets()** എന്ന ഫങ്ഷൻ ഉപയോഗിക്കാവുന്നതാണ്. **gets()** ഫങ്ഷന് വൈറ്റ് സ്പേസ് അടങ്ങിയ സ്ട്രിങ്ങുകൾ സ്വീകരിക്കാൻ കഴിയും.

Input/Output functions to handle strings

(i) gets() function: വൈറ്റ് സ്പെയ്സുകളുള്ള ഒരു സ്ട്രിംഗ് ഇൻപുട്ട് ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്നു.

(i.e, if we input the string "Maya Mohan", the entire string "Maya Mohan" is stored.)
Eg: gets(my_name);

(ii) puts() function: ഒരു സ്ട്രിംഗ് ഔട്ട്പുട്ട് ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്നു.

Eg: puts(my_name);

Chapter 3
Functions

- **Function:** ഒരു പ്രത്യേക ജോലി ചെയ്യുന്ന നാമകരണം ചെയ്യപ്പെട്ട ഒരു കൂട്ടം സ്റ്റേറ്റ്മെന്റുകൾക്ക് **ഫങ്ഷൻ** എന്ന് പറയുന്നു.
- ഫങ്ഷനുകളെ **Predefined functions**, **User-defined functions** എന്നിങ്ങനെ രണ്ടായി തിരിച്ചിരിക്കുന്നു**.**

Predefined functions

- **Predefined functions (built-in functions):** നേരത്തെ നിർവചിക്കപ്പെട്ടിട്ടുള്ളതും സി++ ഹെഡർ ഫയലുകളിൽ സൂക്ഷിക്കപ്പെട്ടിട്ടുള്ളതുമായ ഫങ്ഷനുകൾ.

  Predefined functions are classified into:

1. Console functions for character I/O

ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് **cstdio** എന്ന ഹെഡ്ഡർ ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തണം.

i. getchar() - ഒരു കാരക്ടർ ഇൻപുട്ട് ചെയ്യുന്നതിന്.

Eg:- char ch = getchar();

ii. putchar() - നൽകിയിരിക്കുന്ന കാരക്ടർ സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നതിന്.

```
Eg:-   char ch = 'B';
       putchar(ch);          //displays 'B' on the screen
```

2. **Stream functions for I/O operations**

ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് **iostream** എന്ന ഹെഡ്ഡർ ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തണം.

A) Input functions

i. get(): ഒരു കാരക്ടറോ സ്ട്രിംഗോ ഇൻപുട്ട് ചെയ്യുന്നതിന്.

```
Eg:-   ch=cin.get(ch);              //accepts a character and stores in ch
```

ii. getline(): ഒരു സ്ട്രിംഗ് ഇൻപുട്ട് ചെയ്യുന്നതിന്.

```
Eg:-   cin.getline(str,10);         // accepts a string of maximum 10 characters
```

B) Output functions

i. put(): നൽകിയിരിക്കുന്ന കാരക്ടർ സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നതിന്.

```
Eg:-   char ch='c';
       cout.put(ch);         //character 'c' is displayed
```

ii. write(): നൽകിയിരിക്കുന്ന സ്ട്രിംഗ് സ്ക്രീനിൽ പ്രദർശിപ്പിക്കുന്നതിന്.

```
Eg:-   char str[10]="hello";
       cout.write(str,10);   //dispaly the string “hello”
```

3. **String functions**

സ്ട്രിങ്ങുകൾ കൈകാര്യം ചെയ്യുന്നതിന് ഉപയോഗിക്കുന്നു.

ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് **cstring** എന്ന ഹെഡ്ഡർ ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തണം.

i. strlen(): ഒരു സ്ട്രിംഗിന്റെ നീളം കണ്ടുപിടിക്കുന്നതിന്. (i.e., സ്ട്രിംഗിലെ കാരക്ടറുകളുടെ എണ്ണം).

```
Eg:- n = strlen(“Welcome”);         //Stores 7 in n.
```

ii. strcpy(): ഒരു സ്ട്രിങ്ങിനെ മറ്റൊരു സ്ട്രിങ്ങിലേക്ക് copy ചെയ്യുന്നതിന്.

```
Eg:-   char str[10];
       strcpy(str,"Welcome");       // string constant "Welcome" will be stored in str
```

iii. strcat(): ഒരു സ്ട്രിങ്ങിലേക്ക് മറ്റൊരു സ്ട്രിംഗ് കൂട്ടിച്ചേർക്കുന്നതിന് .

```
Eg:-   char s1[20] = "Welcome", s2[10] = " to C++";
       strcat(s1,s2);
       cout << s1;     // will display "Welcome to C++"
```

iv. strcmp(): രണ്ട് സ്ട്രിംഗുകൾ തമ്മിൽ താരതമ്യം ചെയ്യുന്നതിന്.

```
Eg:-   strcmp(string1, string2)
```

v. strcmpi(): വലിയ അക്ഷരം ചെറിയ അക്ഷരം എന്ന വ്യത്യാസമില്ലാതെ രണ്ട് സ്ട്രിംഗുകൾ തമ്മിൽ താരതമ്യം ചെയ്യുന്നതിന്.

4. **Mathematical functions**

ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് **cmath** എന്ന ഹെഡ്ഡർ ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തണം.

i. abs(): ഒരു പൂർണ്ണസംഖ്യയുടെ absolute value കണ്ടുപിടിക്കുന്നതിന് ഉപയോഗിക്കുന്നു.

```
Eg:-    int n = -25;
        cout << abs(n);        //Returns 25.
```

ii. sqrt(): ഒരു സംഖ്യയുടെ square root കണ്ടുപിടിക്കുന്നതിന് ഉപയോഗിക്കുന്നു.

```
Eg:-    sqrt(25);        //Returns 5.
```

iii. pow(): ഒരു സംഖ്യയുടെ power കണ്ടുപിടിക്കുന്നതിന് ഉപയോഗിക്കുന്നു.

Eg:-   z = pow(4, 3);   //Returns 64. (i.e., $4^3 = 4*4*4 = 64$)

5. Character functions

ഈ ഫങ്ഷനുകൾ ഉപയോഗിക്കുന്നതിന് **cctype** എന്ന ഹെഡ്ഡർ ഫയൽ പ്രോഗ്രാമിൽ ഉൾപ്പെടുത്തണം.

i. isupper(): തന്നിരിക്കുന്ന കാരക്ടർ വലിയക്ഷരത്തിലാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നു. വലിയ അക്ഷരത്തിലാണെങ്കിൽ ഒരു +ve നമ്പറും അല്ലെങ്കിൽ പൂജ്യവും ഫംഗ്ഷൻ return ചെയ്യുന്നു.

```
Eg:-    char c = 'A';
        int n = isupper(c);     //Returns a +ve number
```

ii. islower(): തന്നിരിക്കുന്ന കാരക്ടർ ചെറിയക്ഷരത്തിലാണോ അല്ലയോ എന്ന് പരിശോധിക്കുന്നു. ചെറിയ അക്ഷരത്തിലാണെങ്കിൽ ഒരു +ve നമ്പറും അല്ലെങ്കിൽ പൂജ്യവും ഫംഗ്ഷൻ return ചെയ്യുന്നു.

```
Eg:-    char c = 'A';
        int n = islower(c);     //Returns 0
```

iii. isalpha(): തന്നിരിക്കുന്ന കാരക്ടർ alphabet ആണോ എന്ന് പരിശോധിക്കുന്നു. alphabet ആണെങ്കിൽ ഒരു +ve നമ്പറും അല്ലെങ്കിൽ പൂജ്യവും ഫംഗ്ഷൻ return ചെയ്യുന്നു.

```
Eg:-    int n = isalpha('3');     //Returns 0
        int n = isalpha('a');     //Returns a +ve number
```

iv. isdigit(): തന്നിരിക്കുന്ന കാരക്ടർ digit ആണോ എന്ന് പരിശോധിക്കുന്നു. digit ആണെങ്കിൽ ഒരു +ve നമ്പറും അല്ലെങ്കിൽ പൂജ്യവും ഫംഗ്ഷൻ return ചെയ്യുന്നു.

```
Eg:-    char c = 'b';
        int n = isdigit(c);              //Returns 0
```

v. isalnum(): തന്നിരിക്കുന്ന കാരക്ടർ alphabet, digit എന്നിവ ആണോ എന്ന് പരിശോധിക്കുന്നു. alphabet or digit ആണെങ്കിൽ ഒരു +ve നമ്പറും അല്ലെങ്കിൽ പൂജ്യവും ഫംഗ്ഷൻ return ചെയ്യുന്നു.

```
Eg:-    int n = isalnum('3');     //Returns a +ve number
        int n = isalnum('a');     //Returns a +ve number
        int n = isalnum('@');     //Returns 0
```

vi. toupper(): തന്നിരിക്കുന്ന കാരക്ടർ വലിയക്ഷരത്തിലേക്ക് മാറ്റുന്നു.

```
Eg:-    char c = toupper('a');          //Returns A
```

vii. tolower(): തന്നിരിക്കുന്ന കാരക്ടർ ചെറിയക്ഷരത്തിലേക്ക് മാറ്റുന്നു.

```
Eg:-    char c = tolower('A');          //Returns a
```

Methods of calling functions

- ഒരു ഫംഗ്ഷനിലേക്ക് ആർഗ്യുമെന്റുകൾ രണ്ട് തരത്തിൽ കൈമാറാൻ കഴിയും.

1. **Call by value (Pass by value) method**
   ഈ രീതിയിൽ, യഥാർത്ഥ ആർഗ്യുമെന്റിന്റെ ഒരു പകർപ്പ് ഫംഗ്ഷനിലേക്ക് അയക്കുന്നു.
2. **Call by reference (Pass by reference) method**
   ഈ രീതിയിൽ, യഥാർത്ഥ ആർഗ്യുമെന്റിന്റെ റഫറൻസ് (അഡ്രസ്സ്) ഫംഗ്ഷനിലേക്ക് അയക്കുന്നു.

Call by value v/s Call by reference

Call by Value Method	**Call by Reference Method**
സാധാരണ വേരിയബിളുകൾ formal parameter കളായി ഉപയോഗിക്കുന്നു.	റഫറൻസ് വേരിയബിളുകൾ formal parameter കളായി ഉപയോഗിക്കുന്നു.
formal argument കളിൽ വരുത്തിയ മാറ്റങ്ങൾ actual argument കളിൽ പ്രതിഫലിക്കുന്നില്ല.	formal argument കളിൽ വരുത്തിയ മാറ്റങ്ങൾ actual argument കളിൽ പ്രതിഫലിക്കുന്നു.

Chapter 4
Web Technology

- ഒരു വെബ് പേജ്, കമ്പ്യൂട്ടർ/ മൊബൈൽ ബ്രൗസെറിലൂടെ ലോകത്ത് എവിടെ നിന്നും access ചെയ്യാൻ സാധിക്കും
- HTML ടാഗുകളും അതിന്റെ attributes ഉം ആണ് അടിസ്ഥാനപരമായി വെബ് പേജ് നിർമ്മാണത്തിന് ഉപയോഗിക്കുന്നത്.

Static v/s Dynamic web pages

Static web page

- സ്റ്റാറ്റിക് വെബ് പേജിൽ എപ്പോൾ സന്ദർശിക്കുമ്പോഴും ഒരേ content ആണ് കാണുക. reading ആവശ്യത്തിന് ഉപയോഗിക്കുന്നു.

Dynamic (interactive) web page

- input സ്വീകരിക്കാനും, അതനുസരിച്ച് content മാറുകയും ചെയ്യുന്ന വെബ്പേജുകൾ

Static	Dynamic
content, layout തുടങ്ങിയവ മാറ്റമില്ലാതെ നിലനിൽക്കുന്നു	Runtime ൽ content, layout എന്നിവ മാറുന്നു
database ഉപയോഗിക്കുന്നില്ല	database ഉപയോഗിക്കുന്നു
നിർമ്മിക്കാൻ എളുപ്പമാണ്	programming പഠിച്ചവർക്ക് മാത്രമേ dynamic പേജ് ചെയ്യാൻ സാധിക്കൂ

script

HTML പേജിൽ എഴുതുന്ന പ്രോഗ്രാം കോഡിനെയാണ് സ്ക്രിപ്റ്റ് എന്ന് വിളിക്കുന്നത്.

അതിന് വേണ്ടി ഉപയോഗിക്കുന്ന Tag <script>

Client side Script v/s Server side Script	
Client side Script	**Server side Script**
script കോഡ് client ബ്രൗസറിലേക്ക് കോപ്പി ചെയ്യപ്പെടുന്നു	script എല്ലായ്പ്പോഴും web സെർവറിൽ ആയിരിക്കും
data validation ന് പ്രധാനമായി ഉപയോഗിക്കുന്നു	database ഉപയോഗിക്കാനും web server ൽ നിന്ന് data അയക്കാനും.
client script, user ന് block ചെയ്യാൻ സാധിക്കും	server side scripting, യൂസർ ന് block ചെയ്യാൻ സാധിക്കില്ല.

Eg:

client script: Java script, VB script,
server side script: PHP, JSP, ASP, Perl

Basic Structure of HTML Page

```
<HTML>
        <HEAD>
                <TITLE>Title of the Page</TITLE>
        </HEAD>
        <BODY>
                All Webpage content here...
        </BODY>
</HTML>
```

HTML Tags & Attributes

Tag – HTML command. Angle bracket (< >) ന്റെ ഉള്ളിൽ എഴുതുന്നു.

Attribute-ടാഗുകളുടെ additional information നെയാണ് attribute എന്ന് വിളിക്കുന്നത്.

Container Tag & Empty Tag

- opening (< >), closing </ >ഉള്ള ടാഗുകളെയാണ് container Tag എന്ന് വിളിക്കുന്നത്.

  ഉദാ:<HTML> </HTML>, <BODY> </BODY>

- Closing വേണ്ടാത്ത ടാഗുകളെയാണ് Empty Tag എന്ന് വിളിക്കുന്നത്.

  ഉദാ:

HTML Tags

Tags	Use	Attributes	Values & Purpose
<HTML>	To Start an HTML document	Dir, Lang	

<HEAD>	To specify head Section of the web page		
<TITLE>	Tile to be displayed on Title bar of browser		
<BODY>	Defines the body section of the web page	Bgcolor	To set back ground color of web page. Color Name or Hexadecimal color code can be given as value
		Background	To set an image as back ground of web page
		Text	To set color of text in web page
		Link, Alink,Vlink	To set color of hyperlinks

Tags	Use	Attributes	Values & Purpose
<H1>...................... <H6>	Different Levels of Headings. <H1> biggest and <H6> smallest	Align	"left", "right" and "center" are the values
 	To break current line and continue in the next line		
<P>	To create a paragraph		
<HR>	To draw horizontal line	Size	thickness of line
		Width	to decide length of line
		Align	alignment of line
		Color	
<CENTER>	<CENTER> </CENTER> ടാഗിന്റെ ഉള്ളിൽ എഴുതുന്നതെന്തും center alignment ൽ ആയിരിക്കും		

Text Formatting Tags

Tags	Use	Attributes	Values & Purpose
<B> and <STRONG>	To make a text **Bold** Face		
<I> and <EM>	To make a text *italics* or emphasis		
<U>	To underline the text		
<S> and <STRIKE>	To ~~strike through~~ the text		
<BIG>	To make a text big sized		

<SMALL>	To make a text small sized		
<SUB>	To make a text subscript		
<SUP>	To make a text superscript		
<Q>	To enclose the text in double quotes		
<BLOCKQUOTE>	To indent the text		
<PRE>	tag defines pre formatted text. The text will be displayed exactly as written in the HTML source code		
<ADDRESS>	tag defines the contact information for the author		
<DIV>	tag defines a division or a section in an HTML document.		

Tags	**Use**	**Attributes**	**Values & Purpose**
<MARQUEE>	To scroll a text or image in a browser	Height	set height of marquee in pixels
		Width	specify the width of marquee to display.
		Direction	which direction the marquee to scroll
		Loop	number of times the marquee to appear in web page
		Bgcolor	specifies the back ground color
<FONT>	To change the size, style, colour of the text enclosed	color	To set the text color
		font	It specifies the font face like Arial, Verdana etc.
		size	Font size
<IMG>	To insert an image in a web page	Src	To specify the file name of the image

Chapter 5
Web Page Designing using HTML

- ഒരു വെബ് പേജിൽ പല തരത്തിലുള്ള ലിസ്റ്റുകൾ ഉപയോഗിക്കാം. അവയിൽ പ്രധാനപ്പെട്ടത് താഴെ കൊടുക്കുന്നു
  - **Unordered List**
  - **Ordered List**
  - **Definition List**

Unordered List (Bullet list)

- ഇതിന് വേണ്ടി ഉപയോഗിക്കുന്ന container tag <UL> </UL>
- ലിസ്റ്റിലെ ഓരോ ഐറ്റവും <LI> </LI> ഉപയോഗിച്ച് എഴുതുന്നു.

Eg:

```
<UL>
	<LI>RAM</LI>
	<LI>CPU</LI>
	<LI>MOTHER BOARD</LI>
	<LI>PROCESSOR</LI>
</UL>
```

Output

- RAM
- CPU
- MOTHER BOARD
- PROCESSOR

<UL> ന്റെ പ്രധാനപ്പെട്ട attribute ആണ് Type. ലിസ്റ്റിനെ പല രൂപത്തിൽ ആക്കാൻ ഉപയോഗിക്കുന്നു. അതിന്റെ values square, circle, disc എന്നിവയാണ്. അതിൽ default value disc ആണ്.

Ordered List (Numbered List)

This list in numerical or alphabetical order

- ഇതിന് വേണ്ടി ഉപയോഗിക്കുന്ന container tag <OL> </OL>
- ലിസ്റ്റിലെ ഓരോ ഐറ്റവും unordered ലിസ്റ്റ് പോലെ <LI> </LI> ഉപയോഗിച്ച് എഴുതുന്നു.

Eg:

```
<UL>
	<LI>RAM</LI>
	<LI>CPU</LI>
	<LI>MOTHER BOARD</LI>
	<LI>PROCESSOR</LI>
</UL>
```

Output

1. RAM
2. CPU
3. MOTHER BOARD
4. PROCESSOR

Unordered ലിസ്റ്റ് പോലെ Type attribute ഇവിടെയും ഉപയോഗിക്കുന്നു. list customise ചെയ്യാൻ സഹായിക്കുന്നു. Type ന്റെ values താഴെ കൊടുക്കുന്നു.

1 -> 1,2,3,4......
A->A,B,C,D,.....
a->a,b,c,d,......
I->I,II,III,IV,.....
i->i,ii,iii,iv,.....

<OL> ലിസ്റ്റിന്റെ മറ്റൊരു പ്രധാനപ്പെട്ട attribute ആണ് start എന്നത്. ലിസ്റ്റിന്റെ തുടക്കത്തിലെ value മാറ്റുന്നതിന് ഉപയോഗിക്കുന്നു. start ന്റെ value ഉദ്ദേശിക്കുന്നത് starting position ആണ്.

Eg:

<OL TYPE=”I” START=5> will start a list with index value V.

Definition List

It is list of terms and corresponding definitions

- ഇതിന് വേണ്ടി ഉപയോഗിക്കുന്ന container tag <DL> </DL>
- Definition ലിസ്റ്റുമായി ബന്ധപ്പെട്ട മറ്റ് പ്രധാനപ്പെട്ട ടാഗുകൾ <DT>, <DD> എന്നിവയാണ്

Eg:

```
<DL>
        <DT>RAM</DT>
                <DD> RAM is main memory. It is also called Read Write Memory. It
                        is volatile, can not store data permanently
                </DD>
        <DT>HARD DISK</DT>
                <DD> HARD DISK is secondary storage device. Data stored
                        permanently. Hard disk has very high data storage capacity
                </DD>
</DL>
```

Links in HTML

- Hyperlink എന്നത് ഒരു text / image ആണ്. അതിൽ click ചെയ്യുമ്പോൾ പുതിയ പേജിലേക്ക് അല്ലെങ്കിൽ അതേ പേജിന്റെ മറ്റൊരു ഭാഗത്തേക്ക് നീങ്ങാൻ സാധിക്കുന്നു.
- <A> ടാഗ്, anchor tag എന്നും അറിയപ്പെടുന്നു.
- <A> ടാഗിന്റെ പ്രധാനപ്പെട്ട attribute - Href
- Href ന്റെ value വെബ് പേജിന്റെ അല്ലെങ്കിൽ web സൈറ്റിന്റെ അഡ്രസ് ആണ്

Eg:

<A href=”http://www.dhsekerala.gov.in”>Higher Secondary</A>

Internal Linking

- ഒരു വെബ് പേജിന്റെ മറ്റൊരു ഭാഗത്തേക്ക് പോകുവാൻ സഹായിക്കുന്ന link നെയാണ് internal linking എന്ന് വിളിക്കുന്നത്
- <A> ടാഗിന്റെ Name എന്ന attribute അതിന് സഹായിക്കുന്നു.

Eg:

```
<A  Name="Introduction">INTRODUCTION</A>
<A Href="#Introduction">Go To Introduction</A>
```

Go To Introduction എന്ന text ൽ click ചെയ്താൽ cursor, Introduction എന്ന് Name ചെയ്ത ഭാഗത്തേക്ക് പോകും.

External Linking

ഒരു വെബ് പേജിൽ നിന്ന് മറ്റൊരു വെബ് പേജിലേക്ക് ഉള്ള link നെയാണ് External linking എന്ന് പറയുന്നത്.

Table Tags

<TABLE>	**- To create table**
<TR>	**- To create a row**
<TH>	**- To create an heading cell**
<TD>	**- To specify a data in a cell**

- <TABLE> Table content here </TABLE>

<TABLE> ടാഗിന്റെ പ്രധാനപ്പെട്ട attributes താഴെ കൊടുക്കുന്നു

- border – thickness of table border line, if it's value is 0, means no border
- bordercolor – color of border
- align   -ബ്രൌസർ വിൻഡോവിന്റെ എവിടെ വേണം ടേബിൾ എന്ന് തീരുമാനിക്കുന്നു.
- bgcolor-Table ന്റെ back ground colorാറ്റുന്നതിന്

Other Table Tags

- <TR> row contents here</TR> ഒരു row എന്നത് collection of cells ആണ്.
- <TH> Heading cell </TH>
- <TD>Data Cells</TD>

Eg:

```
<TABLE BORDER="1">
        <TR>
                <TH>ROLL NO</TH>
                <TH>NAME</TH>
        </TR>
        <TR>
                <TD>1</TD>
                <TD>Joy</TD>
        </TR>
```

```
        <TR>
            <TD>2</TD>
            <TD>Arun</TD>
        </TR>
    </TABLE>
```

output

Roll No	Name
1	Joy
2	Arun

Attributes of <TR>

- align - ഒരു row യുടെ ഉള്ളിലുള്ള text ഏത് alignment ൽ വേണം (left, right, center)
- valign - vertical alignment of the content in a cell of the row (top,middle,bottom)
- bgcolor -back ground color of the row

Attributes of <TH> and <TD>

- align - ഒരു cell ന്റെ ഉള്ളിലുള്ള text ഏത് alignment ൽ വേണം (left, right, center)
- valign - vertical alignment of the content in a cell (top,middle,bottom)
- bgcolor -back ground color of the cell.

<FORM> Tag

വെബ് പേജ് കാണുന്നവരിൽ നിന്ന് സെർവറിലേക്ക് ഡാറ്റ ശേഖരിക്കുന്നതിന് form ഉപയോഗിക്കുന്നു.

ഉദാ: www.hscap.kerala.gov.in എന്ന വെബ് സൈറ്റിലെ ഫോമുകളിലൂടെയാണ് അഡ്മിഷനാവശ്യമായ വിവരങ്ങൾ ശേഖരിച്ചത്

ഒരു form ൽ പ്രധാനമായും രണ്ട് elements ഉൾക്കൊള്ളുന്നു.

1. <FORM> container tags
2. Number of controls with in the form (text field,radio button, checkbox, etc.)

Attributes of <FORM>

1. Action : URL of Form handler
(form handler എന്ന് പറയുന്നത്, form ൽ നിന്ന് server ൽ എത്തുന്ന Data യെ കൈകാര്യം ചെയ്യാൻ server ൽ ഉള്ള പ്രോഗ്രാമിനെ ആണ്.)

2. Method: It mentions the method used to upload data (Get and Post)

Form controls

- വിവിധ തരത്തിലുള്ള form controls ഉണ്ട് .ആ controls ഫോമിൽ ലഭിക്കുന്നതിന് സഹായിക്കുന്ന tag

I. <INPUT>

- **<INPUT> ടാഗിന്റെ പ്രധാന attribute Type**

Type attribute ന്റെ പ്രധാന values

1) text -> create a textbox
2) password -> Create a password box. That means characters appear as coded symbols
3) checkbox ->Creates a check box
4) radio ->create an option button (user ക്ക് ഒരു സമയത്ത് ഒരു ഗ്രൂപ്പിൽ ഒരു ഓപ്ഷൻ മാത്രമേ select ചെയ്യാൻ സാധിക്കൂ).
5) reset ->To create reset button, clears everything in the form.
6) submit -> submit button
7) button -> creates a standard graphical button

<INPUT> ടാഗിന്റെ മറ്റ് attributes

- Name – to name an input control
- Value – It is used to provide an initial (default) value inside the control
- Size – Width of input text. (Only for text and password)
- maxlength -It limits the number of characters that the user can type into the field. It is also applicable to text & password

---

Chapter 6
Client Side Scripting Using JavaScript

ക്ലൈന്റ് സൈഡ് സ്ക്രിപ്റ്റിംഗിന് ഉപയോഗിക്കുന്ന ഒരു സ്ക്രിപ്റ്റിംഗ് ലാംഗ്വേജ് ആണ് ജാവ സ്ക്രിപ്റ്റ്. Netscape browser ന വേണ്ടി ബ്രണ്ടൻ ഈച് ആണ് ഇത് ഡെവലപ് ചെയ്തത് .ഇത് ഒട്ടുമിക്ക വെബ് ബ്രൗസറുകളിലും ഉപയോഗിക്കാം

<SCRIPT> Tag

HTML പേജിൽ സ്ക്രിപ്റ്റിംഗ് കോഡ് ഉൾപെടുത്താൻ <SCRIPT> tag ഉപയോഗിക്കുന്നു .ഇതിലെ Language attribute ഏതു scripting language ആണ് ഉപയോഗിക്കുന്നത് എന്ന് വ്യക്തമാക്കുന്നു.

Eg:-
```
<SCRIPT Language=”javascript”>
    ...........................................................
    ...........................................................
    </SCRIPT>
```

***Java Script* ഉപയോഗിച്ച് എങ്ങിനെ ഒരു *web page* ഉണ്ടാക്കാം എന്ന് നോക്കാം**

```
<HTML>
<HEAD> <TITLE>Javascript - Welcome</TITLE> </HEAD>
<BODY>
<SCRIPT Language= "JavaScript">
document.write("Welcome to JavaScript.");
</SCRIPT>
</BODY>
</HTML>
```

ഈ program save ചെയ്യുന്നത് **.html** എന്ന extension ൽ ആണ് .ഇവിടെ ഒരു text , HTML page ൽ ഉൾപ്പെടുത്താൻ വേണ്ടിയാണ് **document.write()**എന്ന ജാവാസ്ക്രിപ്റ്റ് കമാൻഡ് ഉപയോഗിച്ചത് .

<u>Data Types</u>

ജാവാസ്ക്രിപ്റ്റിൽ ബേസിക് data type കളെ മൂന്നായി തരം തിരിക്കാം

1.Number 2.String 3.Boolean

Number - ഏല്ലാ വിധത്തി ലുള്ള സംഖ്യ കളെയും സൂചിപ്പിക്കുന്നു

String-Double quotes നള്ളിൽ ഉൾപ്പെടുത്തിയിട്ടുള്ള ഏല്ലാ ക്യാരക്ടറുകളും

Boolean-ഈ data type ൽ ആകെ true , false എന്നി രണ്ടുവാല്യൂസ് ആണ് ഉള്ളത്

<u>Variables in JavaScript</u>

ഡാറ്റകൾ സൂക്ഷിക്കുന്നതിനാണ് variables ഉപയോഗിക്കുന്നത് .ജാവാസ്ക്രിപ്റ്റിൽ variables ഡിക്ലയർ ചെയ്യുന്നതിന് **var** എന്ന **keyword** ആണ് ഉപയോഗിക്കുന്നത്

Syntax : var x; (x എന്നത് variable ന്റെ പേര്).വേരിയബിളുകൾക്ക് വാല്യൂ നൽകുമ്പോൾ മാ ത്ര മാണ് അവയുടെ ഡേറ്റാ ടൈപ്പ് തീരുമാനിക്കപ്പെടുന്നത്

Eg: var x,y;
x=10;
y="KERALA"; ഇവിടെ x എന്ന വേരിയബിളിന്റെ ടൈപ്പ് നമ്പറും y എന്ന വേരിയബിളിന്റെ ടൈപ്പ് സ്ട്രിങ്ങും ആണ് .

<u>Operators in JavaScript</u>

C++ ൽ ഉപയോഗിക്കുന്ന ഓപ്പറേറ്റർസ് അതുപോലെതന്നെ ജാവാസ്ക്രിപ്റ്റിലും ഉപയോഗിക്കുന്നു

Arithmetic Operator is used to perform arithmetic operations . (+, -, *, /, %)
Relational Operator is used for comparison operations . (<, <=, >, >=, ==, !=)
Assignment Operator (=)
Arithmetic Assignment Operator (+=, - =, * =, / =, % =) are used to simplify the use of assignment operator .
Logical Operator – (Logical and &&, Logical or ||, logical not !)

String addition operator(+)

രണ്ടു സ്ട്രിങ്ങുകളെ കൂട്ടിച്ചേർക്കുന്നതിനാണ് ഈ ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നത് .ഇത് C++ ൽ ഇല്ല .

```
Eg: var x,y;
     x="Good";
    y="Morning";
    z=x+y;
```

ഇപ്പോൾ z നകത്തു Good Morning എന്ന വാല്യൂ ആയിരിക്കും ഉണ്ടാകുക .

typeof() operator

ഡേറ്റ ടൈപ്പ് എന്താണെന്നു മനസിലാക്കുന്നതിനാണ് typeof()ഓപ്പറേറ്റർ ഉപയോഗിക്കുന്നത് .

Example :-

```
typeof( " Hello " )// returns String
typeof( 3.14 )// returns Number
typeof( false) // returns Boolean
var p ;
typeof( p )// returns Undefned(കാരണം p ക്ക് വാല്യൂ നൽകിയിട്ടില്ല )
```

Control structures in JavaScript

പ്രോഗ്രാമിന്റെ തുടർച്ചയായ നീക്കത്തിന് മാറ്റമുണ്ടാക്കുന്നതാണ് control structures. C++ ൽ ഉപയോഗിക്കുന്ന control structures തന്നെയാണ് ഇവിടെയും ഉപയോഗിക്കുന്നത് .

1) if statement

```
if ( test _ expression )
{
Statements ;
}
else
{
Statements ;
}
```

2) switch statement

```
switch ( expression )
{
case value 1: statement 1; break ;
case value 2: ststement 2; break ;
-
----------------------
default : statement ;
}
```

3) for Loop

```
for ( initialisation ; expression ; update _ statement )
{
statements ;
}
```

4) while

while (expression)
{
statements ;
}

Built - in Functions

1) **alert ( ) function** - സ്ക്രീനിൽ ഒരു മെസ്സേജ് ബോക്സ് കാണിക്കുവാൻ alert() ഉപയോഗിക്കുന്നു .

Syntax : alert (“ message ”);

2) **isNaN ( ) function** - ഒരു സ്ട്രിങ് നമ്പർ ആണോ എന്ന് പരിശോധിക്കുന്നു .നമ്പർ ആണെങ്കിൽ false ഉം അല്ലെങ്കിൽ true ഉം റിട്ടേൺ ചെയ്യുന്നു

Syntax : isNaN (“ value”);
Eg : isNaN (“65”) ---> False
isNaN (“ Big ”) ---> True

3) **toUpperCase ( ) function** - ഒരു സ്ട്രിങ്ങിനെ uppercase ലേക്ക് മാറ്റുവാൻ ഉപയോഗിക്കുന്നു

Example :-
var a , b ;
a =” abcd ”;
b = a . toUpperCase ( );
document . write (b);
Output : ABCD

4) **toLowerCase ( ) function** - ഒരു സ്ട്രിങ്ങിനെ lowercase ലേക്ക് മാറ്റുവാൻ ഉപയോഗിക്കുന്നു

5) **charAt () function** - ഒരു സ്ട്രിങ്ങിന്റെ പറഞ്ഞിരിക്കുന്ന സ്ഥാനത്തുള്ള character ഏതാണെന്ന് അറിയിക്കുന്നു

Example :-
var s = " HELLO WORLD ";
var r = str . charAt (0);
H റിട്ടേൺ ചെയ്യുന്നു

6)**length property** - സ്ട്രിങ്ങിനകത്തുള്ള characters ന്റെ എണ്ണം റിട്ടേൺ ചെയ്യുന്നു.

Example :-
var x, n;
x = “JavaScript”;
n = x.length;
10 റിട്ടേൺ ചെയ്യുന്നു

6) **Number () function** – ഒരു ന്യൂമെറിക് സ്ട്രിങ്ങിന്റെ വാല്യൂ റിട്ടേൺ ചെയ്യുന്നു

Example : Number (“65”) --> 65

Chapter 7
Web Hosting

വ്യക്തികളോ സ്ഥാപനങ്ങളൊ നിർമ്മിക്കുന്ന വെബ്സൈറ്റുകൾ ഇൻറർനെറ്റിൽ ലഭ്യമാക്കുന്നതിന് വെബ്സെർവറുകളിൽ സ്റ്റോറേജ് സ്പേസ് അനുവദിക്കുന്ന സേവനത്തിനെ **Web Hosting** എന്ന് പറയുന്നു ഈ സേവനങ്ങൾ ലഭ്യമാക്കുന്ന കമ്പനികളെ **web hosts** എന്ന് പറയുന്നു .

Types of web hosting

Web hosting നെ മൂന്നായി തരം തിരിക്കാം

1.Shared hosting
2.Dedicated hosting
3.Virtual Private Server

1-Shared Hosting

ഇത് ഏറ്റവും സാധാരണമായ web hosing ആണ് .ഒന്നിലധികം വെബ് സൈറ്റുകൾ ഒരു സെർവറിൽ ഹോസ്റ്റ് ചെയ്യുകയും ആ സെർവറിന്റെ RAM ,CPU മുതലായവ പങ്കുവെക്കുകയും ചെയ്യുന്നതിനാലാണ് ഇതിനെ shared hosting എന്ന് പറയുന്നത് .കുറച്ച് സന്ദർശകറുള്ള ചെറിയ വെബ്സൈറ്റുകൾക്ക് shared hosting അനുയോജ്യമാണ് .ഇത് മറ്റു ഹോസ്റ്റിംഗിനേക്കാൾ ചെലവ് കുറഞ്ഞതാണ് .

2-Dedicated Hosting

Dedicated hosting ൽ ഒരു web server ഉം അനുബന്ധ ഉപകരണങ്ങളും ഒരു web site നു വേണ്ടി മാത്രം ഉപയോഗിക്കുന്നു .കൂടുതൽ സ്റ്റോറേജ് സ്പേസ് ,കൂടുതൽബാൻഡ് വിഡ്ത് എന്നിവ ആവശ്യമുള്ളതും കൂടുതൽ സന്ദർശകരുള്ളതുമായ വെബ് സൈറ്റുകൾക്ക് dedicated hosting അനുയോജ്യമാണ് .ഇത് ഏറ്റവും ചെലവുകൂടിയ ഹോസ്റ്റിങ് രീതിയാണ് .

3.Virtual Private Server

Shared hosting നേക്കാൾ കൂടുതൽ സൗകര്യങ്ങൾ ആവശ്യമുള്ളതും എന്നാൽ dedicated hosting ന്റെ അത്ര സൗകര്യങ്ങൾ ആവശ്യമില്ലാത്തവർക്കും വേണ്ടിയുള്ള ഹോസ്റ്റിങ് രീതിയാണ് Virtual Private Server(VPS).ഇവിടെ ഓരോ വെബ്സൈറ്റിനും സ്വന്തമായി RAM,band width,OS എന്നിവ ലഭ്യമാണ് .Dedicated hosting നേക്കാൾചെലവ് കുറഞ്ഞതും എന്നാൽ shared hosting നേക്കാൾ ചെലവ് കൂടിയതുമായ ഹോസ്റ്റിങ് രീതിയാണ് VPS.

FTP client software

ഒരു വ്യക്തിയുടെ കമ്പ്യൂട്ടറിനെ വെബ്സെർവറുമായി ബന്ധപ്പെടുത്തി വെബ് പേജ് ലഭ്യമാക്കുന്നതിന് FTP client software ഉപയോഗിക്കുന്നു .FTP server നെ കണക്ട് ചെയ്യാൻ FTP client software ന് ഒരു യൂസർ നേമും പാസ്സ്‌വേർഡും ഡൊമൈൻനേമും ആവശ്യമാണ് .യൂസർ നേമും പാസ്സ്‌വേർഡും പ്ലെയിൻ ടെക്സ്റ്റ് ആയാണ് സെർവറിലേക്ക് അയക്കുന്നത് .ഇതിനു സെക്യൂരിറ്റി കുറവാണ് .അതിനാൽ ഇപ്പോൾ SSH FTP(SFTP) protocol ഉപയോഗിച്ച് യൂസർ നേമും പാസ്സ്‌വേർഡും ഡാറ്റയും എൻ ക്രിപ്റ്റ് ചെയ്താണ് സെർവറിലേക്ക് അയക്കുന്നത് .FileZilla, CuteFTP, SmartFTP എന്നിവയാണ് പ്രസിദ്ധമായ FTP client software കൾ.

Free hosting

വ്യക്തികൾക്കോ സ്ഥാപനങ്ങൾക്കോ പൈസമുടക്കാതെ വെബ് ഹോസ്റ്റിങ് സാധ്യമാക്കുന്ന സേവനമാണിത് .വെബ് സൈറ്റു കളിൽ പ്രദർശിപ്പിക്കുന്ന പരസ്യങ്ങളിൽ നിന്നാണ് സേവനദാതാക്കൾ ഇതിനുവേണ്ട ചെലവ് കണ്ടെത്തുന്നത്.മറ്റുള്ള ഹോസ്റ്റിംഗിനേക്കാൾ കുറഞ്ഞ സേവനങ്ങളെ free hosting ൽ ലഭ്യമാകൂ .വളരെ കുറഞ്ഞ സൈസുള്ള ഫയലുകൾ മാ ത്ര മേ അപ്‌ലോഡ് ചെയ്യാൻ അനുവദിക്കൂ .കൂടാതെ ഓഡിയോ, വീഡിയോ ഫയലുകൾ അനുവദിക്കാറില്ല

ഡൊമൈൻ നെയിം ലഭിക്കുന്നതുനു ഫ്രീ ഹോസ്റ്റിങ് കമ്പനികൾ രണ്ടു തരത്തിലുള്ള സേവങ്ങളാണ് നൽകുന്നത്

1. Directory Service :- കമ്പനി അഡ്രസ് / നമ്മുടെ അഡ്രസ് (www.example.com/oursite)

2.Sub Domain :- നമ്മുടെ അഡ്രസ് .കമ്പനി അഡ്രസ് (oursite.example.com)

Chapter 8
Database Management System

Database is an organized collection of inter-related data stored together with minimum redundancy, which can be retrieved as desirable.

Database Management System (DBMS) is essentially a set of programs which facilitates storage, retrieval and management of database.

Advantages of DBMS

a). Controlling Data redundancy:-Duplication of data is known as data redundancy. In DBMS data is kept in one place in centralized manner and the users can access this centrally maintained date for their purpose. So data redundancy can control.

b). Data consistency :- Data redundancy leads data inconsistency (different copies of same data hold different values because the updation of data may not occur in all the copies. In DBMS it avoided by eliminating redundancy.

c). Efficient data access: DBMS utilizes a variety of techniques to store and retrieve data efficiently.

d). Data can be shared:- The data stored in the Database can be shared among many users and new programs can be developed to share the existing data.

e). Data Integrity: Integrity refers to the overall completeness, accuracy and consistency of data in the database. It can be achieved by use of error checking, validation, avoiding duplication etc.

f). Security :- Information inside a Database is very valuable for a company. Security refers to accidental or intentional disclosure or unauthorized access, modification or destruction. Through the use of passwords, information in the database is made available only to authorized person. Access to specific information can be limited to selected users by using access rights.

g). Sharing of data : The data stored in the database can be shared among several users or programs even simultaneously and each may use it for different purposes.

h). Enforcement of Standards:- The database Administrator defines standards like display formats, report structure, update procedures, access rules etc.. for the DBMS. It is helpful when data transfer occurs between systems.

i). Crash recovery : If the system crashes data in the database may become unusable. DBMS provides some mechanism to recover data from the crashes.

Components of DBMS

a). Hardware : It include actual computer system used for storage and retrieval of Database. It include computers, storage devices, network devices, and other supporting devices.

b). Software : It consists of DBMS, application programs and utilities. DBMS consists of different components that handles tasks like data definition, data manipulation,

c). Data : It is the most important component of DBMS. The data base should contain all the data needed by the organization. For effective storage and retrieval of information , data is organized as fields, records and files.

1) Field : A field is a smallest unit of stored data. Each field has a specific type.(eg. Name,-Char, Mark,-num etc).

2) Record : A record is a collection of related fields. A record store an instant of the relation or the table.

3) File : A file is a collection of all occurrence of one type of record.

d). Users : The users of database can be classified depending on the mode of their interactions with DBMS. The different categories of users are Database Administrator, Application Programmer, Sophisticated users, and naïve users.

e). Procedures : Procedures refers to the instructions and rules that govern the design and use of the database.

Database users

Users of the database can be classified in to the following groups,

1. Database Administrator (DBA) : The person who is responsible for the control of the centralized and shared database is the Database Administrator(DBA).
2. Application Programmers : Application programmers are computer professionals who interact with the DBMS through applicationprograms
3. Sophisticated Users : Sophisticated users include engineers, scientists, business analysts, and others who are thoroughly familiar with the facilities of theD B M S .
4. Naive Users: Naive users interact with the system by invoking one of the application programs that were written previously. They are not even aware of the details of the DBMS.

RDBMS terminologies

a). Entity : An entity is a person or a thing that can be distinguishable from others.
Eg. Student, Book etc..

b). Relation : Relation is a collection of data elements organized in rows and columns. It is also called a table.

c). Tuple : A row in a relation is called a tuple. It consists of a complete set of values used to represent a particular entity. Values of each attribute are taken from its domain.

d). Attribute : The columns of a relation are called attributes.
Eg. Ad.no, name, age etc are the attributes of the relation Student.

e). Degree : The no of attributes(columns) in a relation is called degree of that relation.
f). Cardinality : The no of rows (tuples) in a relation is called cardinality of that relation
g). Domain :Domain is a range of values from which a particular value of an attribute for a can be drawn.
Eg. 0 to 100 may be domain of values for the attribute mark in student relation.
h). Instance : An instance of a relation is a set of tuples has the same number of fields as the relational scheme.

Relational operations

The collection of operations that is used to manipulate the entire relations of a database is known as relational algebra. These operations are performed with the help of a special language, called query language. The fundamental operations in relational algebra are SELECT, PROJECT, UNION, INTERSECTION, SET DIFFERENCE, CARTESIAN PRODUCT, etc. The SELECT and PROJECT operations are unary operations because they operate on one relation. The remaining operations are binary operations as they operate on pairs of relations.
1) Select operation: Select operation is a unary operation used to select tuples in a relation those satisfy a selection predicate (condition). It is denoted by the symbol σ
2) Project operation: Project operation is a unary operation used to select from a table to form a new relation. It is denoted by the symbol ∏.
3) Union Operation: It is the concatenation operation between compactable (same type) where all the tuples appearing in either or both the relations.. It is denoted by U.
Duplication of tuples are eliminated.
4). Intersection Operation: This binary operation done between compactable relations. It returns a relation consisting of all tuples appearing I both of the two specified relations. It is denoted by the symbol ∩ .
5) Set Difference Operation: This binary operation done between compactable relations. It returns a relation consisting of all tuples appearing in the first relation and not in the second relation. It is denoted by the symbol − .
6)Cartesian Product(Cross Product): This operation returns a relation consisting of all possible combinations of tuples from two relations. It is a binary operation on two relations which has degree equal to the sum of the degrees of the two relations and cardinality is the product of the cardinalities of the two relations. It is denoted by X.

Chapter- 9
Structured Query Language (SQL)

Structured Query Language (SQL) is a language designed for managing data in RDBMS. It provides facilities to create a table, insert data into a table, retrieve information from a table, modify data in the table, delete the existing data from a table, modify the structure of a table, remove a table from a database, etc.

Components of SQL

Data Definition Language (DDL), Data Manipulation language (DML) and Data Control Language (DCL).

DDL commands are used to create, modify and remove the database objects such as tables, views and keys. Eg: CREATE TABLE, ALTER TABLE, DROP TABLE, CREATE VIEW, DROP VIEW.
DML permits users to insert data into tables, retrieve existing data, delete data from tables and modify the stored data. Eg: INSERT INTO, SELECT, UPDATE, DELETE FROM .
DCL includes commands that control a database, including administering privileges and committing data. Eg: GRANT, REVOKE .

SQL Data Types

SQL Data types are : INT or INTEGER, DEC or DECIMAL, CHAR or CHARACTER, VARCHAR , DATE , TIME.
INT or INTEGER is used to define columns contain integer data.
DEC(5,2) or DECIMAL(5,2) denotes that the column with this specification can store any value having a maximum of five digits, out of which two are after the decimal point.
CHAR is a fixed length character data type. It is mainly used when the data in a column are of the same fixed length and small in size. VARCHAR represents variable length strings. The space allocated for the data depends only on the actual size of the string, not on the declared size of the column.

Constraints

Constraints are rules applied on data entered into the column of a table. They ensures the integrity of the relation . So they are known as integrity constraints.
Constraints may be table Constraints or Column Constraints. A table constraint is applied to a table where as a column constraint is applied to a column.

1) **NOT NULL** -This constraint ensures that a column can never have NULL(empty) values.
2) **AUTO_INCREMENT** -The AUTO_INCREMENT keyword perform an auto increment ie, it automatically assigns a series of number automatically and insert it to column. The default starting value is 1.The auto increment column must be defined as primary key of the table. Only one auto_increment column is allowed in a table.
3) **UNIQUE** - This constraint ensures that no two rows have the same value in a specified column. This constraint can be applied to those columns that have been declared NOT NULL.

4) **PRIMARY KEY** - It declares a column as the primary key of a table. This column must not have null values and every value should be unique. So a PRIMARY KEY constraint is a combination of UNIQUE and NOT NULL constraints.
5) **DEFAULT** - This constraint is used to specify a default value for a column.
6) **CHECK** - This constraint limits the values that can be inserted into a column of a table.

SQL commands

DDL commands

Command	Use	Syntax
CREATE TABLE	To create a table	CREATE TABLE tbl_name (col_name data type <constraint>, col_name data type <constraint>, ... col_name data type <constraint>);
DROP TABLE	To remove a table	DROP TABLE <table_name>;

DML commands

Command	Use	Syntax
SELECT	To retrieve information from the table	SELECT col_name1, col_name2,... FROM tbl_name;
INSERT INTO	To insert a record	INSERT INTO table_name VALUES(value1,value2,value3);
UPDATE	To modify the column values	UPDATE table_name SET col_name=value [WHERE condition];
DELETE	To delete rows / records	DELETE FROM table_name [WHERE condition]

Chapter 10
ERP (Enterprise Resource Planning)

ERP ഒരു കമ്പനിയുടെ എല്ലാ ആവശ്യകതകളും സംയോജിപ്പിച്ച് ഒരു central database ലേക്ക് സംയോജിപ്പിക്കുന്നതിനാൽ വിവിധ Departments തമ്മിൽ വിവരങ്ങൾ പങ്കിടാൻ കഴിയും.

Functional units of ERP

Financial Module:
ഈ Module വിവിധ വകുപ്പുകളിൽ നിന്ന് financial data ശേഖരിക്കുകയും Balance sheet, Trial balance, General ledger മുതലായ വിവിധ Reports സൃഷ്ടിക്കുകയും ചെയ്യുന്നു.

Manufacturing Module:
ഈ Module മുഴുവൻ production process ന് ആവശ്യമായ Information Manage ചെയ്യുകയും നൽകുകയും ചെയ്യുന്നു.

Production Planning Module:

Production പരമാവധിയാക്കാനും നഷ്ടം കുറയ്ക്കാനും Optimize ചെയ്ത രീതിയിൽ Resources വിനിയോഗത്തിനായി ഈ module ഉപയോഗിക്കുന്നു. Optimization എന്നാൽ Production വേണ്ടി Resources ഫലപ്രദമായി ഉപയോഗിക്കുക എന്നതാണ്.

HR Module: Personal details, salary details, attendance and promotion തുടങ്ങിയ വിവരങ്ങൾ അടങ്ങിയ Update ചെയ്തതും പൂർണ്ണവുമായ Employee details database HR module പരിപാലിക്കുന്നു.

Inventory Control module: ഈ Module ഒരു Organization ന്റെ Stock requirement നിയന്ത്രിക്കുന്നു.

Purchasing Module: ഈ Module, Raw materials ശരിയായ സമയത്ത് ശരിയായ വിലയ്ക്ക് ലഭിക്കുന്നതിന് സഹായിക്കുന്നു. Suppliers, billing മുതലായവയ്ക്കായി Purchase orders സൃഷ്ടിക്കുന്നു.

Marketing Module: ഈ module customer orders നിരീക്ഷിക്കുന്നതിനും customer satisfaction വർദ്ധിപ്പിക്കുന്നതിനും credit risks ഇല്ലാതാക്കുന്നതിനും ഉപയോഗിക്കുന്നു.

Sales and Distribution Module: ഈ module , sales and distribution activities നിയന്ത്രിക്കുന്നു. അന്വേഷണങ്ങൾ, Order placement തുടങ്ങിയവയും ഈ Module ൽ ഉൾപ്പെടുന്നു. ഈ Module, e-commerce വെബ്സൈറ്റുമായി സംയോജിപ്പിക്കുന്നു.

Quality Management Module: Quality Planning, Quality Inspection ,Quality Control എന്നിവ ഈ Module കൈകാര്യം ചെയ്യുന്നു.

Examples for ERP packages

Oracle – Oracle ൽ നിന്നുള്ള ERP package ശക്തമായ finance and accounting module നൽകുന്നു. മികച്ച customer and supplier interaction, effective production analysis, efficient human resource management , better pricing module എന്നിവയും നൽകുന്നു.

SAP - SAP എന്നാൽ Systems, Applications and Products for data processing. Customer Relationship Management (CRM), Supply Chain Management (SCM), Product Life cycle Management (PLM) software എന്നിവയും SAP develop ചെയ്തു.

Odoo – Open source ERP package. Open ERP എന്ന് ആണ് ആദ്യനാമം.

Microsoft Dynamics - Microsoft business solutions ന്റെ ഭാഗമാണ് Microsoft Dynamics. ഇത് പ്രധാനമായും ഇടത്തരം സംരംഭങ്ങളെ ലക്ഷ്യം വച്ചുള്ള ഒരു കൂട്ടം ERP ഉൽപ്പന്നങ്ങൾ നൽകുന്നു. ഈ Package Install ചെയ്യാനും എളുപ്പത്തിൽ ഉപയോഗിക്കാനും കഴിയും. കൂടാതെ മികച്ച user interface നൽകുന്നു.

Tally ERP - Accounting, inventory payroll എന്നിവക്ക് വേണ്ടിയുള്ള business accounting software ആണ് Tally ERP.

Benefits of ERP

1. Improved resource utilization: ERP install ചെയ്യുന്നത് Resources ന്റെ പാഴാക്കൽ കുറയ്ക്കുകയും resource utilization മെച്ചപ്പെടുത്തുകയും ചെയ്യും.

2. Better Customer Satisfaction: Customer satisfaction എന്നാൽ ഒരു ഉൽപ്പന്നത്തിനോ സേവനത്തിനോ Customers requirements പരമാവധി തൃപ്തിപ്പെടുത്തുക എന്നതാണ്. Web based ERP അവതരിപ്പിക്കുന്നതിലൂടെ ഒരു ഉപഭോക്താവിന് വീട്ടിൽ നിന്ന് തന്നെ ഓർഡറുകൾ നൽകാനും Payments നടത്താനും കഴിയും

3. Provides Accurate Information:-ERP software ഉപഭോക്താവിന് manual systems നെക്കാൾ accurate information നൽകുന്നു.

4. Decision Making Capability:-കൃത്യവും പ്രസക്തവുമായ വിവരങ്ങൾ Systems ന് മികച്ച തീരുമാനങ്ങൾ എടുക്കാൻ സഹായിക്കുന്നു.

5. Increased Flexibility:-ERP systems ന് പുതിയ മാറ്റങ്ങൾ എളുപ്പത്തിൽ പൊരുത്തപ്പെടുത്താൻ കഴിയും.

6. Information Integrity:-Enterprise നെക്കുറിച്ചുള്ള മുഴുവൻ വിവരങ്ങളും Central Database store ചെയ്തിരിക്കുന്നതിനാൽ ഏത് Department നും information നൽകാൻ കഴിയും.

Chapter-11
Trends and issues in ICT

Mobile communication services

Short Message Service (SMS) 160 അക്ഷരങ്ങൾ വരെ ഉൾക്കൊള്ളുന്ന ഹ്രസ്വ വാചക സന്ദേശങ്ങൾ കൈമാറാൻ അനുവദിക്കുന്ന മൊബൈൽ ആശയവിനിമയ സംവിധാനമാണ് SMS.

Multimedia Messaging Service (MMS) മൊബൈൽ ഫോണുകൾ ഉപയോഗിച്ച് Multimedia ഉള്ളടക്കം അടങ്ങുന്ന സന്ദേശങ്ങൾ അയയ്ക്കുന്നതിനും സ്വീകരിക്കുന്നതിനുമുള്ള ഒരു standard മാർഗമാണ്. ഒരു Multimedia സന്ദേശത്തിനായി MMS, maximum size വ്യക്തമാക്കുന്നില്ല, മാത്രമല്ല ഇത് text, graphics, music, video clips എന്നിവ പോലുള്ള ഉള്ളടക്കങ്ങളെ പിന്തുണയ്ക്കുന്നു.

Global Positioning System (GPS) ഒരു Satellite അധിഷ്ഠിത Navigation സംവിധാനമാണ്. ഭൂമിയിലെവിടെയും longitude and latitude ഉപയോഗിച്ച് ഭൂമിശാസ്ത്രപരമായ സ്ഥാനം കണ്ടെത്താൻ സഹായിക്കുന്നു. കമ്പനികളുടെ ട്രക്കുകളുടെ ചലനം Track ചെയ്യുന്നതിന് ജിപിഎസ് വാഹനങ്ങളുടെ Tracking നായി ഉപയോഗിക്കുന്നു.

Mobile Operating System

ഒരു മൊബൈൽ ഉപകരണത്തിലെ hardware, multimedia functions, Internet connectivity, തുടങ്ങിയവ കൈകാര്യം ചെയ്യുന്ന Software ഉദാഹരണം: Android from Google, iOS from Apple, BlackBerry OS from BlackBerry and Windows Phone from Microsoft.

Android mobile OS

പ്രധാനമായും Touch screen മൊബൈൽ ഉപകരണങ്ങളായ smart phones , tablet computers എന്നിവയ്ക്കായി design ചെയ്തിരിക്കുന്ന Linux അടിസ്ഥാനമാക്കിയുള്ള operating system ആണ്. ആൻഡി റൂബിനും കൂട്ടുകാരും ചേർന്ന് 2003 ൽ കാലിഫോർണിയയിലെ Palo Alto യിൽ സ്ഥാപിച്ച Android Inc. ആണ് ഇത് ആദ്യം വികസിപ്പിച്ചെടുത്തത്. 2005 ൽ Google, Android Inc സ്വന്തമാക്കി പൂർണ ഉടമസ്ഥതയിലുള്ള ഉപസ്ഥാപനമാക്കി.

Cyber Crime

കമ്പ്യൂട്ടറുകളോ കമ്പ്യൂട്ടർ നെറ്റ്‌വർക്കുകളോ ഒരു ഉപകരണം, target അല്ലെങ്കിൽ criminal activity യുടെ സ്ഥലമായി ഉപയോഗിക്കുന്ന ഒരു ക്രിമിനൽ പ്രവർത്തനമായാണ് ഇത് നിർവചിച്ചിരിക്കുന്നത്. സൈബർ കുറ്റകൃത്യങ്ങളിൽ phishing, hacking, denial of service attacks തുടങ്ങിയവ ഉൾപ്പെടുന്നു.

Cyber crimes against individuals

Identity theft -

ഒരു വ്യക്തിയുടെ അനുവാദമില്ലാതെ പേര്, ക്രെഡിറ്റ് കാർഡ് നമ്പർ മുതലായ തിരിച്ചറിയുന്ന വിവരങ്ങൾ വഞ്ചനയ്ക്കോ മറ്റ് കുറ്റകൃത്യങ്ങൾക്കോ ഉപയോഗിക്കുക.

Harassment – Chat rooms, Social media , e-mail മുതലായവയിൽ വ്യക്തികൾക്കു നേരെ ലിംഗഭേദം, വംശം, മതം, ദേശീയത എന്നിവ കേന്ദ്രീകരിച്ച് അപമാനകരമായ അഭിപ്രായങ്ങൾ പോസ്റ്റുചെയ്യുന്നത്.

Impersonation and cheating - ഇരയെ ദ്രോഹിക്കുന്നതിനായി മറ്റൊരു വ്യക്തിയായി ആൾമാറാട്ടം നടത്തുക.

Violation of privacy – തക്കതായ കാരണമില്ലാതെ മറ്റൊരാളുടെ സ്വകാര്യ ജീവിതത്തിലേക്കു കടന്നു കയറി സ്വകാര്യത ലംഘിക്കുക.

Dissemination of obscene material - Pornography പോലുള്ള കുറ്റകൃത്യങ്ങൾ നടത്തുന്നതിന് ഇന്റർനെറ്റ് ഒരു മാധ്യമം ആക്കുക. ഇന്നത്തെ പ്രധാന സൈബർ കുറ്റകൃത്യങ്ങളിലൊന്നാണ് അശ്ലീല വസ്തുക്കളുടെ വിതരണവും പോസ്റ്റുചെയ്യലും.