CRITERION 6 - FACILITIES AND TECHNICAL SUPPORT

6.2 Additional facilities created for improving the quality of learning experience in laboratories (25)

S.No	Document Name	Available
1	Lab manual	Yes
2	Knowledge Wall	Yes
3	LCD projector	Yes
4	Internet Facility	Yes

6.2 Additional facilities created for improving the quality of learning experience in laboratories (25)

Sr.No	Facility Name	Details	Reason(s) for creating facility	Utilization	Areas in which students' are expected to have enhanced learning	Relevance to POs/PSOs
1	Lab Manual	Provide all practical Lab Manual to student	Practical Guidance	Students	All subjects	PO1,PO2,PO3,PO4,PO5, PO6,PO7,PO8,/PSO1, PSO2
2	Knowledge Wall	Subject Model	Basic idea of subject	Student	Basic Knowledge of subjects	PO1,PO2,PO3,PO4, PO6,PO9,PO10, PO11/PSO1, PSO2
3	LCD projector	ICT tool	Presentation	Student and Faculties	Seminar, Workshop, Value added course	PO1,PO2,PO3,PO10, PO11/PSO1, PSO2
4	Internet Facility	Wi-Fi	Providing high speed connectivity	Student and Faculties	Project, Seminar, Subjects	PO1,PO2,PO3,PO7,PO8,PO9,PO10, /PSO1, PSO2

Lab Manual

srivenkateshwaraa
College of Engineering & Technology
ASPIRE TO EXCEL

ISO 21001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DESIGN AND ANALYSIS OF ALGORITHMS LABORATORY

CS P42

LAB MANUAL

LIST OF EXPERIEMENTS

1. Implementation of binary search using Divide-and-Conquer technique.
2. Implementation of merge sort algorithms using Divide-and-Conquer technique.
3. Implementation of quick sort algorithms using Divide-and-Conquer technique.
4. Implementation of Knapsack using Greedy technique.
5. Implementation of Single-Source Shortest Paths algorithms using Greedy techniqu
6. Implementation of Multi-Stage Graphs using Dynamic Programming technique.
7. Implementation of 0/1 Knapsack using Dynamic Programming technique.
8. Implementation of All Pairs Shortest Paths using Dynamic Programming techniqu
9. Implementation of Traveling Salesman algorithms using Dynamic Programming technique.
10. Implementation of Pre-order, In-order, Post-order traversals using DFS travers techniques.
11. Implementation of 8 Queens with the design of Backtracking.
12. Implementation of sum of subsets with the design of Backtracking.
13. Implementation of 0/1 Knapsack problems with Branch-and-Bound technique.
14. Implementation of Traveling Salesman problems with Branch-and-Boun technique.

Expt. No: 1 **IMPLEMENTATION OF BINARY SEARCH**

AIM

To write a C++ program to implement Binary search using divide and conquer.

ALGORITHM

Step 1: Start the program.

Step 2: Declare the variables and an array of elements.

Step 3: Read the value of number of elements to be stored in the array.

Step 4: Read the values of the elements of the array.

Step 5: Divide the array of elements into two which gives a middle element.

Step 6: Get the element to be searched in the array.

Step 7: Compare the element with the middle element. If the element to be searched is the middle element stop searching and print the position.

Step 8: If the element is lesser than the middle element the searching proceeds with the first half of the array.

Step 9: If the element is greater than the middle element the searching proceeds with the second half of the array.

Step 10: If the element is not present in the array printl Unsuccessful searchl.

Step 11: If the element is present in the array, print the position of the searched element.

Step 12: Stop the program.

PROGRAM

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
class binary
{
int i,mid,low,high,n,x,a[20];
public:
void get();
int search();
};
void main()
{
int index;
binary  b;
clrscr();
b.get();
index = b.search();
if(index != -1)
{
```

```
cout<<"\nElement found in "<< index+1 <<" position";
}
else
{
cout<<"\nElement not found ";
}
getch();
}
void binary::get()
{
cout<<"\nEnter the number of elements :"; cin>>n;
cout<<"\nEnter the array elements:\n";
for(i=0;i<n;i++) {
cin>>a[i];
}
cout<<"\nEnter search elements:\n"; cin>>x;
}
int binary::search()
{
low=0;high=n-1; while (low<=high)
{
mid=(low+high)/2;
if(x==a[mid])
{
return mid;
}
else if(x>a[mid])
{
low=mid+1;
}
else if(x<a[mid])
{
high=mid-1;
}
}
return -1;
}
```

OUTPUT

```
Enter the number of elements :5
Enter the array elements:
12
26
10
5
2
Enter search elements:
1
Element not found
```

RESULT

Thus, the program for binary search using divide and conquer was executed and verified successfully.

srivenkateshwaraa
College of Engineering & Technology
A SHINE TO EXCEL

(Approved by AICTE | Accredited By NAAC 'A' Grade | Affiliated to Pondicherry University)
(An ISO 21001:2018 Certified Institution)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

Course & Dept. Name	:	B.Tech. & CSE
Year & Semester	:	II & II
Subject Name	:	DATA STRUCTURES AND ALGORITHMS
Subject Code	:	CSPL302

CSPL302 DATA STRUCTURE AND ALGORITHMS LAB

L T P C 0 0 4 2

Course Pre-requisite:

- Basic knowledge in programming

Course Objective:

- To enable students write programs using various data structures, analyse and understand the benefits of choosing the right data structure.

Course Outcomes:

- To write programs for search and sorting algorithms.
- To write programs for implementing stacks, queues and linked list.
- To write programs for searching using tree data structure.
- To write programs for identifying shortest path in a network.
- To write programs that implements hash tables.

LIST OF EXPERIMENTS

1. Searching Algorithms (With the Number of Key Comparisons) - Sequential, Binary and Fibonacci Search Algorithms on an Ordered List
2. Sorting Algorithms: Insertion Sort, Selection Sort, Bubble Sort, Quick Sort, Heap Sort and Merge Sort.
3. Implementation of Stack and Its Operations.
4. Application of Stack for Converting an Arithmetic Expression into Postfix Form and Evaluation of Postfix Expression.
5. Implementation of Queue, Circular Queue, Priority Queue, Dequeue and Their Operations.
6. Implementation of Singly Linked List, Doubly Linked List, Circular Linked List.
7. Implementation of Binary Tree and Binary Traversal Techniques.
8. Implementation of Graph Traversal Techniques.
9. Implement Dijkstra's Algorithm to Obtain the Shortest Paths.
10. Implementation of Hash Tables and its Operations.

Ex. No:1 **Searching Algorithms (With the Number of Key Comparisons) - Sequential, Binary and Fibonacci Search Algorithms on an Ordered List**
Date:

1a. Sequential Search:

Aim: To perform sequential search of an element on the given array.

Algorithm:

1. Start
2. Read number of array elements n
3. Read array elements Ai, i = 0,1,2,...n−1
4. Read search value
5. Assign 0 to found
6. Check each array element against search
7. If Ai = search then
found = 1
Print "Element found"
Print position i
Stop
8. If found = 0 then
print "Element not found"
Stop

Program

```
/* Linear search on a sorted array */
#include <stdio.h>
#include <conio.h>
main()
{
nt a[50],i, n, val, found;
lrscr();
rintf("Enter number of elements : ");
canf("%d", &n);
rintf("Enter Array Elements : \n");
```

```
for(i=0; i<n; i++)
scanf("%d", &a[i]);
printf("Enter element to locate : ");
scanf("%d", &val);
found = 0;
for(i=0; i<n; i++)
{
if (a[i] == val)
{
printf("Element found at position %d", i);
found = 1;
break;
}
}
if (found == 0)
printf("\n Element not found");
getch();
}
```

Output

```
Enter number of elements : 7
Enter Array Elements :
23 6 12 5 0 32 10
Enter element to locate : 5
Element found at position 3
```

Result

Thus an array was linearly searched for an element's existence.

CS P43 OBJECT ORIENTED PROGRAMMING LABORATORY

LIST OF EXPERIMENTS

1. Program to implement classes and objects.
2. Program to implement constructors and destructors with array of objects.
3. Program to demonstrate function overloading.
4. Program to implement different types of inheritances like multiple, Multilevel and hybrid.
5. I/O Program to demonstrate the use of abstract classes.
6. Program to demonstrate I/O streams and functions.
7. Program to perform all possible type conversions.
8. Program to demonstrate exception handling technique.
9. Program to implement networking concepts.
10. Program to implement RMI concepts.
11. Program to implement AWT concepts.
12. Program to implement swing concepts.
13. Program to design and implement applet.
14. Program to design and implement JDBC
15. Program to design an event handling event for simulating a simple calculator.

Ex. No: 1 **CLASSES AND OBJECTS**

AIM:

To write a C++ program to generate Employee Pay Slip using Class & Objects

ALGORITHM:

1. Create a class for the Employee with its data member and member function
2. Get number of employee's n from the user
3. Create object e for the class employee
4. Form a loop for n employees
5. Get the details of the employees
6. Calculate gross pay and net pay
7. Display the employee details
8. Repeat the step 5 until the condition i>n fails
9. Stop the process

CODING:

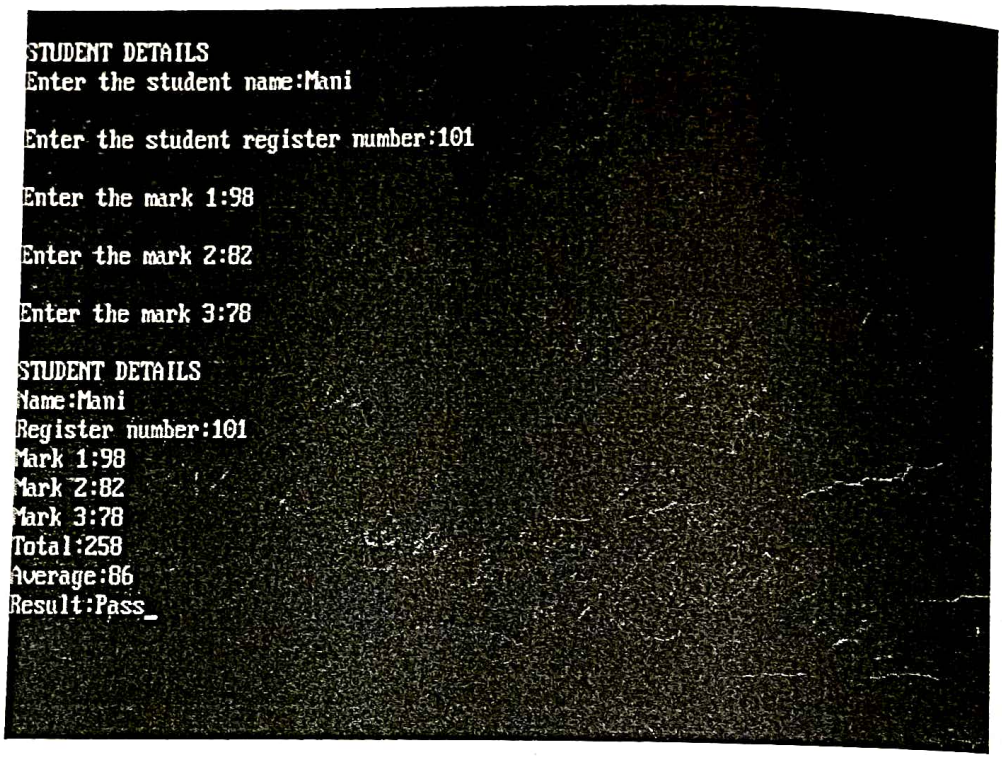
```
#include<iostream.h>
#include<conio.h>
class employee //Class Declaration
{
private:
char ename[20],designation[15]; //Data Member Declaration
float basic,hra,da,ta,pf,gp,net;
public:
void getdetails() //Function to get employee details
{
cout<<"\n\tEnter the Employee name\t\t\t:";
cin>>ename;
cout<<"\n\tEnter the Designation of the Employee :";
cin>>designation;
cout<<"\n\tEnter the Basic pay\t\t\t:";
cin>>basic;
}
void calculation() //Function to calculate salary
{
da= 0.4*basic;
hra=0.25*basic;

 pf=0.1*basic;

ta=0.5*basic;
gp=basic+da+hra+ta;
net=gp-pf;
}
void display() //Function to display employee details
{
cout<<"\n\t\t\tEMPLOYEE DETAILS";
cout<<"\n-----------------------------------------------------";
cout<<"\n\tEmployee name\t:"<<ename;
cout<<"\n\tEmployeeDesig :"<<designation;
```

```
cout<<"\n\tBasic pay\t:"<<basic;
cout<<"\n\tDA\t\t:"<<da;
cout<<"\n\tHRA\t\t:"<<hra;
cout<<"\n\tTA\t\t:"<<ta;
cout<<"\n\tPF\t\t:"<<pf;
cout<<"\n\tGross pay\t:"<<gp;
cout<<"\n\tNetpay\t\t:"<<net;
cout<<"\n-------------------------------------------------";
}
};
void main()
{
clrscr();
employee e; //Object declaration
e.getdetails();
e.calculation();
e.display();
getch();
}
```

OUTPUT:



RESULT:

Thus the program class and objects executed and verified successfully

srivenkateshwaraa
COLLEGE OF ENGINEERING & TECHNOLOGY
(Approved by AICTE, New Delhi & Affiliated to Pondicherry University, Puducherry)
13-A, Pondy-Villupuram Main Road, Ariyur, Puducherry-605 102.
Phone: 0413-2644426, Fax: 2644424 / Website: www.svcetpondy.in

ISO 21001

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

SUBJECT NAME : OPERATING SYSTEM LABORATORY

SUBJECT CODE : CS P51

SEMESTER / YEAR : V / III

HOD

DEAN

PRINCIPAL

LIST OF EXPRIMENTS

1. Study of basic UNIX/Linux commands.
2. Shell Programming.
3. Programs using the following system calls of UNIX/Linux operating system:
   i. fork, exec, getpid, exit, wait, close, stat, opendir, readdir.
4. Programs using the I/O system calls of UNIX operating system:
   i. open, read, write, etc).
5. Simulations of UNIX/Linux commands like ls, grep, etc.
6. Simulation of processes scheduling algorithms.
7. Simulation of synchronization problems using Semaphore.
8. Simulation of basic memory management schemes.
9. Simulation of virtual memory management schemes.
10. Simulation of disk scheduling algorithms
11. Simulation of file systems.
12. Develop an application using any RTOS.

17/6/19

17/6/19
HOD/CSE

.NO.1
TE:

STUDY OF OS LINUX COMMANDS

To study the basic commands in Linux.

MMANDS:

Calendar

NAME : calendar
(i) SYNTAX : cal
DESCRIPTION : Displays a simple calendar. If arguments are not Specified, the current month is displayed.
EXAMPLE : cal
OUTPUT :

June 2014

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

(ii) SYNTAX : cal year
DESCRIPTION : Displays calendar of that year
EXAMPLE : cal 2012
OUTPUT :

January

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

February

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

March

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

April

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

May

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

June

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

July

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

August

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

September

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

October

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

November

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

December

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

(iii) SYNTAX : cal -3
DESCRIPTION : Displays calendar of previous, current, next months of current year
OUTPUT :

July

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

August

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

September

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

(iv)SYNTAX : cal month year
DESCRIPTION : Displays the calendar for corresponding month and year .
EXAMPLE : cal 4 2012
OUTPUT :

April 2012

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

2. Date

NAME : DATE- print or set the system date and time
(i) SYNTAX : date
DESCRIPTION : Display the current time in the given format or set the system date.
OUTPUT : Mon Jul 23 12:17:50 IST 2012

(ii) SYNTAX : date +% H
DESCRIPTION : Display the current hour.
OUTPUT : 12

(iii) SYNTAX : date +% h
DESCRIPTION : Display the current month name.
OUTPUT : Jul

(iv) SYNTAX : date +% m
DESCRIPTION : Display the current month number.
OUTPUT : 7

(v) SYNTAX : date +% a
DESCRIPTION : Display the abbreviated weekday name.
OUTPUT : Mon

(vi) SYNTAX : date +% y
DESCRIPTION : Display the current year.
OUTPUT : 12

(vii) SYNTAX : date +% S
DESCRIPTION : Display the current second.
OUTPUT : 57

Script
NAME : SCRIPT – makes typescript of terminal session
DESCRIPTION : Makes a typescript of everything printed on your terminal. It is useful for students who need a hardcopy record of an interactive session as proof of an assignment, as the typescript file can be printed out later with lpr(1).
SYNTAX : script scriptname
.....
.....
.....
Exit

ENING A SCRIPT:
SYNTAX : vi scriptname
EXAMPLE : vi date.txt
OUTPUT : ~date
~ Mon Jul 23 12:17:50 IST 2012
~
~
INSERT

NAME : LIST – list directory contents
(i) SYNTAX : ls
DESCRIPTION : List information about the Files (the current directory by default).
OUTPUT :

```
greatest.sh    cse.txt        mouse.txt      digit.sh
emp.sh         num.sh         case.sh
```

(ii) SYNTAX : ls –l
DESCRIPTION : Displays files in long listing format.
OUTPUT :

```
csea08    csea08    1384962    Jul 23 12:17 [23.07.2012]
csea08    csea08    20325      Jul 23 12:30 [23.07.12]
csea08    csea08    138        Jul 23 12:33 [01;32case.sh]
csea08    csea08    13830      Jul 23 12:37 [01;34cse]
csea08    csea08    183        Jul 23 12:40 [01;32mdigit.sh]
csea08    csea08    530        Jul 23 12:50 [01;32memp.sh]
csea08    csea08    730        Jul 23 12:59 [01;32mgreatest.sh]
```

(iii) SYNTAX : ls -r
DESCRIPTION : Displays the files in reverse sorted order.
OUTPUT :

```
num.sh         mouse.txt      greatest.sh    emp.sh
digit.sh       cse.txt        case.sh
```

(iv) SYNTAX : ls -s
DESCRIPTION : Displays the size of each files.
OUTPUT :

```
8 num.sh    4 mouse.txt    8 greatest.sh  8 emp.sh
8 digit.sh  4 cse.txt      8 case.sh
```

(v) SYNTAX : ls -S
DESCRIPTION : Displays the files in sorted order.
OUTPUT :

```
case.sh        cse.txt        digit.sh       emp.sh
greatest.sh    mouse.txt      num.sh
```

5. cp

NAME : cp – copy files and directories
SYNTAX : cp fi f2
DESCRIPTION : Copies f1 to f2
EXAMPLE : cp cse.txt new.txt

6. rm

NAME : rm – remove files
SYNTAX : rm filename
DESCRIPTION : This command removes each specified file.
EXAMPLE : rm cse.txt

7. mv

NAME : mv – move(rename) files
SYNTAX : mv f1 f2
DESCRIPTION : Renames Source to Destination
EXAMPLE : mv new.txt cse.txt

mkdir

NAME : mkdir – makes directory
SYNTAX : mkdir DirectoryName
DESCRIPTION : Creates the directory, if they do not exist already
EXAMPLE : mkdir new

rmdir

NAME : rmdir – removes directory
SYNTAX : rmdir DirectoryName
DESCRIPTION : Removes the directory, only it is empty.
EXAMPLE : rm cse.txt

Pwd

NAME : pwd – Present Working Directory displays the name of the current/working directory
SYNTAX : pwd
DESCRIPTION : Displays the name of the current/working directory
OUTPUT : \home\csea08\new

Cd

NAME : cd – Change Directory
(i) SYNTAX : cd dirname
DESCRIPTION : Change the directory which we use to work with.

EXAMPLE : cd New

(ii) SYNTAX : cd ..
DESCRIPTION : Quits from the current directory.

(iii) SYNTAX : cd\
DESCRIPTION : Returns to the home directory.

Cat

NAME : cat- concatenate & open files and print on the standard output
(i) SYNTAX : cat > filename
DESCRIPTION : This command is used to open a new file.
EXAMPLE : cat > a.txt

OUTPUT :

NAME : ZZZZ
ROLL NO: XX

(ii) SYNTAX : cat filename
DESCRIPTION : To view the contents of the file.
EXAMPLE : cat a.txt
OUTPUT :

NAME : ZZZZ
ROLL NO: XX

(iii) SYNTAX : cat f1 f2 > f3
DESCRIPTION : To concatenate f1 and f2 save in f3
EXAMPLE : cat a.txt b.txt > c.txt
OUTPUT :

a.txt=>
NAME : ZZZZ
ROLL NO: XX

b.txt=>
COLLEGE.SVCET

c.txt=>
NAME : ZZZZ
ROLL NO: XX
COLLEGE.SVCET

(iv) SYNTAX : cat –n filename
DESCRIPTION : To display the contents of the file along with the line numbers.
EXAMPLE : cat -n sample.txt
OUTPUT : 1 hai
2 how are u?

(v) SYNTAX : cat f1 >> f2
DESCRIPTION : To redirect the data from one file to another.
EXAMPLE : cat sample.txt new.txt
OUTPUT : cat new.txt

sample. txt=>
hai
how are u?

new. txt=>
hai
how are u?

13. Whoami

NAME : Displays the current user login and identity.
SYNTAX : whoami
OUTPUT : csea08

EX NO: 1 **STUDY OF HTML**

AIM:

To study about the basic **HTML** tags.

HTML:

Hyper Text Markup Language is how a web browser displays its multimedia documents. The documents themselves are plain text files (ASCII) with special "tags" or codes that a browser knows how to interpret and display on your screen.

HTML Tags :

a) Basics Tags:-

i.<html></html>

The basic tag for every page. This tells the browser that the file being loaded is a HTML document.

ii.<head></head>

Head - defines the head of your page. Includes the **<title></title>** tag.

iii. <title></title>

Title - allows you to display a title at the top of the browser.

iv. <Meta>

Meta Tags - allows the owner to display certain information to the browser without the page seeing it. Here are some examples:
<meta name="description" content="This is my page description.">- describe your page
<meta name="keywords" content="word1, word2, word3, word4"> - enter keywords for your page.
<meta HTTP-EQUIV="refresh" content="10; url=index2.html"> - reloads the page after 10 seconds to index2.html
<meta HTTP-EQUIV="text" content="This is my page description."> - another text tag

v. <body></body>

Allows you to define the body arguments. This can include:

- background="file.gif"
- bgcolor="#rgbcode"
- te-xt="#rgbcode"
- link="#rgbcode"

- vlink="#rgbcode"
- topmargin=*n* - defines the top of the margin for the body.
- leftmargin=*n*
- rightmargin=*n*

b) Text Control and Tags:-

i. <h1></h1>

Header - this allows you to change the size of the letter or words it's surrounding. Covers H1 - H6, H1 being the biggest and H6 being the smallest.

**ii. **

Center - allows you to display the text in the center of the page.

iii. <ins></ins>

Inserted Text - allows you to insert text.

iv. <person></person>

Person's Name - allows you to distinguish someone's name, like Shpank.

v. <q></q>

Quotation - set certain test as a quote.

vi. <big></big><small></small>

Big - makes the test bigger than the rest.

Small - makes the text smaller than the rest.

vii.

Subscript - allows you to make the text look like $_{\text{this}}$.

**viii. **

Superscript - gives $^{\text{superscript}}$ effect to your text.

ix. <abbrev></abbrev>

Abbreviation - abbreviate certain text.

x.<del></del>

Deleted Text

**xi. **

Font - allows you to control different aspects of the text.

Includes: -
size=*n* (+1 - +5) (-1 - -5)
color=*#rgbcode* - defines the color
face=*"name"* - defines the font face. Could be Helvetica, Arial, etc.

**xii. **

Bold - makes a word or group of words **bold**. <b&gtbold.

**xiii. **

Strong - basically the same as **bold**, just longer code. <strong&gtbold

**xiv. **

Italics - italicizes a word or *group of words*.

**xv. **

Emphasis - basically the same as *italics*.

xvi. <u></u>

Underline - underlines a word or group of words. But does not work with all browsers.

xvii. <tt></tt>

Typewriter Type - makes a fixed width font.

xviii. <address></address>

Address – another italics tag

xix. <blockquote></blockquote>

Block Quote - indents the left and right-hand sides of the text.

xx. <dfn></dfn>

Definition - allows emboldening or italicizing a word or *group of words*.

xxii. <kbd></kbd>

Keyboard - another fixed width font.

xxiii. <var></var>

Italics - another way to italics word or *group of words.*

xxiv. <pre></pre>

Preformatted - allows the text to appear in the browser as it does on the page.

Lists:-

<dl></dl>

Descriptive List - another way to list things.

<dt></dt>

Defines the topic of the descriptive list.

<dd></dd>

The Descriptive Description. This indented part that is displayed.

Ordered List - a way to group items into a list.

Line item tag defines the list items with a number or a dot. Includes the following:

- start=*n*
- type="A/a/I/i" for upper or lower case and Roman numerals.

Unordered List - another way to list items. Also uses the tags to define the list items with a bullet instead of a number. Includes the following options:

- type="DISC/CIRCLE/SQUARE"

Page Breaks and Lines :-

<hr>

Horizontal Rule - allows you to divide a page with a line. Includes the following options:

- width=pixels/percentage
- align=left/right/center
- size=*n* noshade - takes away the shading inside.

Break - allows the text to break without a full paragraph. The options are:

- clear=left/right/all

<nobr></nobr>

No Breaks - allows the text to continue without breaking.

Frames, Tables, and Forms :-

<table></table>

Tables - these tags allow you to insert tables into your page like all of mine. It has the following options:

- width=*n*
- height=*n*
- border=*n*
- cellpadding=*n*
- cellspacing=*n*

It also includes the following tags with their associated options:

- <tr></tr>
- <td></td>
- <th></th>
- align=left/middle/right
- valign=top/middle/bottom
- color=#rgbcode
- colspan=*n*
- rowspan=*n*

<frameset></frameset>

Frameset - allows you to setup frames on your page. Includes the following:

- rows="pixels/percentage"
- cols="pixels/percentage"
- frameborder=*n*
- framewidth=*n*
- marginheight=*n*
- marginwidth=*n*

Also includes the frame tag to establish content. These include:

- marginheight=*n*
- marginwidth=*n*
- name="name"
- noresize
- src="file.html/file.gif/file.jpg"
- scrolling="yes/no/auto"

The **<noframes>** tag for those browsers who can't handle frames or they won't get bunk.

<frame>

Frame - includes the frame tag to establish content. These include:

- marginheight=*n*
- marginwidth=*n*
- name="name"
- noresize
- src="file.html/file.gif/file.jpg"
- scrolling="yes/no/auto"

<form></form>

Form - this allows you to insert forms onto your page. It includes the following options:

- method=POST/GET
- action="file/script

Also, you can use the following to add checkbox's, text box's, and more. They include:

- <input type="text/hidden/checkbox/radio/submit/reset size="*n*" maxlength="*n*" name="name" value="file/URL">
- <select name="name" size="*n*">
- <option value="value1"&gtValue1</option>

<multicol></multicol>

Multicolumns - allows almost the same effect as tables. It has the following options:

1. cols="*n*"
2. gutter="*n*"
3. width="pixels/percentage"

Images and Links:-

****

Hypertext Anchor - this allows you to make certain text or picture a link to another page or graphic not on the page. It can include the following options:

- target="frame name"

<img>

Image - this allows you to insert a .jpg or .gif image into your web page. It has the following options:

- src="file.gif"/src="file.jpg"
- height=*n*
- width=*n*
- lowsrc="file.gif"/lowsrc="file.jpg"
- alt="text"
- usemap="#mapname"

- ismap
- align="left/right/middle/bottom/top/absmiddle/textop/absbottom"
- border=*n*

<map></map>

Image Map - used to define the areas and coordinates of an image map.

Applets, JavaScript, and Other Tags :-

<applet></applet>

Java Applet - allows you to insert a Java applet directly into your page. It includes:

- code="java.class"
- codebase="/dir/to/applet"
- height="*n*"
- width="*n*"

<script></script>

Javascript - allows a JavaScript to load within the page. This tag usually appears after the </title> and before the </head> tag.

<title>Title</title>
<script language="JavaScript">
Script goes here.
</script>
</head>

<!-- text -->

Allows text to appear invisible on the page. This is used if you would like to save comments on the page, but not have them load on the page.

<marquee></marquee>

Marquee - an IE only tag, so only IE users will see it. It allows scrolling text in the browser. Includes:

- behavior=slide/alternative
- width=pixels/percentage
- hspace=*n*
- vspave=*n*
- loop=n/-1/infinite
- bgcolor=#rgbcode
- scrollamount=*n*
- scrolldelay=*n*

RESULT:

Thus the basic **HTML** tags were studied.

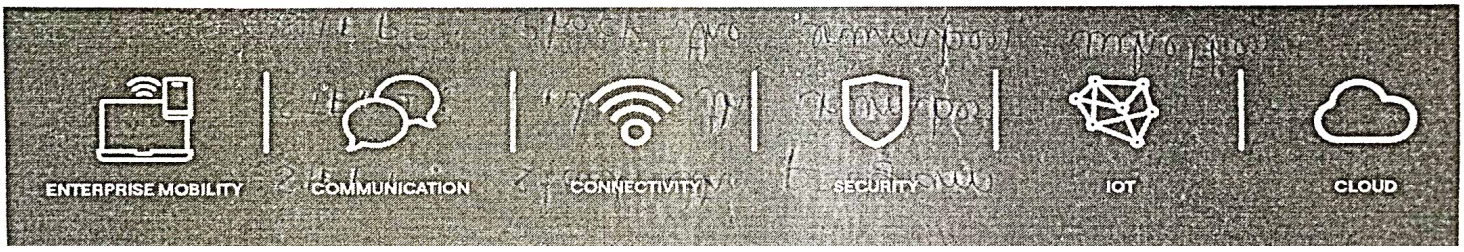
Internet Facility


GPS Map Camera
Pangur, Puducherry, India
Mariamman Koil St, Pangur, Puducherry 605102, India
Lat 11.901791° Long 79.703043°
04/03/2025 11:47 AM GMT +05:30
Google

Your usage details

Company Name	: . SRI VENKATESHWARAA MEDICAL COLLEGE HOSPITAL AND	**Invoice Date**	: 01.07.24
		Invoice Number	: EIPY062400002242
PO Number	: Vi 001 001	**Relationship Number**	: 49495682
PO Date	: 13.03.24	**Circuit ID**	: ENT32CHNPQY102111
Control Number	: 2068578	**Port Bandwidth**	: NA
Plan Name	: ILL_QRC_A2A_Service Charges_INR	**CIR Bandwidth**	: 1024 Mbps
Product Flavor	: BUSINESS INTERNET LEASED LINES(1:1 ILL)	**Annual Charges Service Rental (INR)**	: 1,200,000.00
Billing Periodicity	: Quarterly	**Installation Address**	: PONDY TO VILLUPURAM MAIN ROAD 13A ARIYUR PONDICHERRY 605102 PONDICHERRY

Billing Details for Vi ILL Service: 2068578				**Amount (INR)**
Recurring Charges	**B/W (In Mbps)**	**ARC**	**Charges for the period**	
Service Rental Charges	1024	1200000	19.06.24 to 30.06.24	40,000.00
Service Rental Charges	1024	1200000	01.07.24 to 30.09.24	300,000.00
Sub total				**340,000.00**
Total Rental Charges				**340,000.00**
One time charges				**Net Charges (INR)**
ILL OTC				10,000.00
Total One time charges				**10,000.00**
Tax				**(INR)**
State GST @ 9.00%				31,500.00
Central GST @ 9.00%				31,500.00
Total taxes				**63,000.00**
Total Charges for Vi ILL Service: 2068578				413,000.00



TAX INVOICE

Vi business

your Vi bill

Invoice No: EIPY062400002242

Ship To :
SRI VENKATESHWARAA MEDICAL COLLEGE HOSPITAL AND
RAMACHANDRAN B
13A, PONDY TO VILLUPURAM MAIN ROAD, ARIYUR,
PONDICHERRY - 605102
ARIYUR, PONDICHERRY - 605102
PONDICHERRY - 605102
City: PONDICHERRY
Pin: 605102
State: PONDICHERRY
Kind Attn: RAMACHANDRAN B
Description of service: Telecommunications
Contact No: 9943944669
Nature of Service: ILL

Bill cycle date: 01.07.24

Bill To :
SRI VENKATESHWARAA MEDICAL COLLEGE HOSPITAL AND
RAMACHANDRAN B
13A, PONDY TO VILLUPURAM MAIN ROAD, ARIYUR,
PONDICHERRY - 605102
ARIYUR, PONDICHERRY - 605102
PONDICHERRY - 605102
City: PONDICHERRY
Pin: 605102
State: PONDICHERRY
GSTIN/GSTIN_ISD/UIN No: -
Place of Supply (State): Puducherry
State Code: 34

(details on page 5)

Happy to help

Email us at vibusinessbillingsupport.in@vodafoneidea.com
Call on 180012155666 (Vi toll free)
or +91 9920055666 (Chargeable)

Amount due:INR 413,000.00

Due date:15.07.2024

Relationship no:49495682

Summary of charges for this bill period

	(INR)
• One time charges	10,000.00
• Recurring charges	340,000.00
• Usage charges	0.00
Total value of services	**350,000.00**
• Misc. credits / charges	0.00
Total taxable charges	**350,000.00**
(+) Tax	**63,000.00**
TOTAL PAYABLE	**413,000.00**

Amount in words:
Four Lakh Thirteen Thousand Rupees

Invoice Ref No:

Invoice Date: 01.07.24

No Tax is payable under reverse charge
GST shall be applicable for invoices generated on or after 1st July 2017.

PAN No: AAACB2100P
Vodafone Idea GSTIN: 34AAACB2100P1Z2
HSN Code: 998413
Your previous outstanding balance in (INR): 0.00

Terms & Conditions: 1. Payment not made within due date will carry interest as per agreement. 2. All disputes are subject to Mumbai Jurisdiction only. 3. For Invoice related enquiries, kindly send mail to: backoffice.in@vodafoneidea.com or contact your account manager. 4. For termination request, kindly send mail along with relevant circuit details to :FLTermination.helpdesk@vodafoneidea.com

Vodafone Idea Limited (Formerly Idea Cellular Limited) An Aditya Birla Group & Vodafone Partnership(CIN-L32100GJ1996PLC030976)
Business Office Address: PSA FORT, Plot Nos.A1, A2 & B, Nehru Nagar 1st Main Road, Perungudi, Chennai - 600096
Regd Office Address: SumanTower, Plot No 18, Sector no 11, Gandhinagar 382011, Gujarat -Tel + 91 79 6671 4000

Tear off this slip and return it with your payment. Be sure not to staple.

Payment Slip:

Relationship number: 49495682 **Invoice number:** EIPY062400002242 **Invoice date:** 01.07.24 **Due date:** 15.07.2024 **Amount due INR:** 413,000.00

Cheque/DD in payment of Invoice should be drawn in favour of "Vodafone Idea Limited".
Kindly mention the Relationship number while making the payment, this will facilitates us to allocate the payment against the respective Invoice.

By cheque:

Cheque No:_____
Dated: _____
Bank Name:_____
Bank branch address:_____

PAN No: AAACB2100P HSN Code: 998413

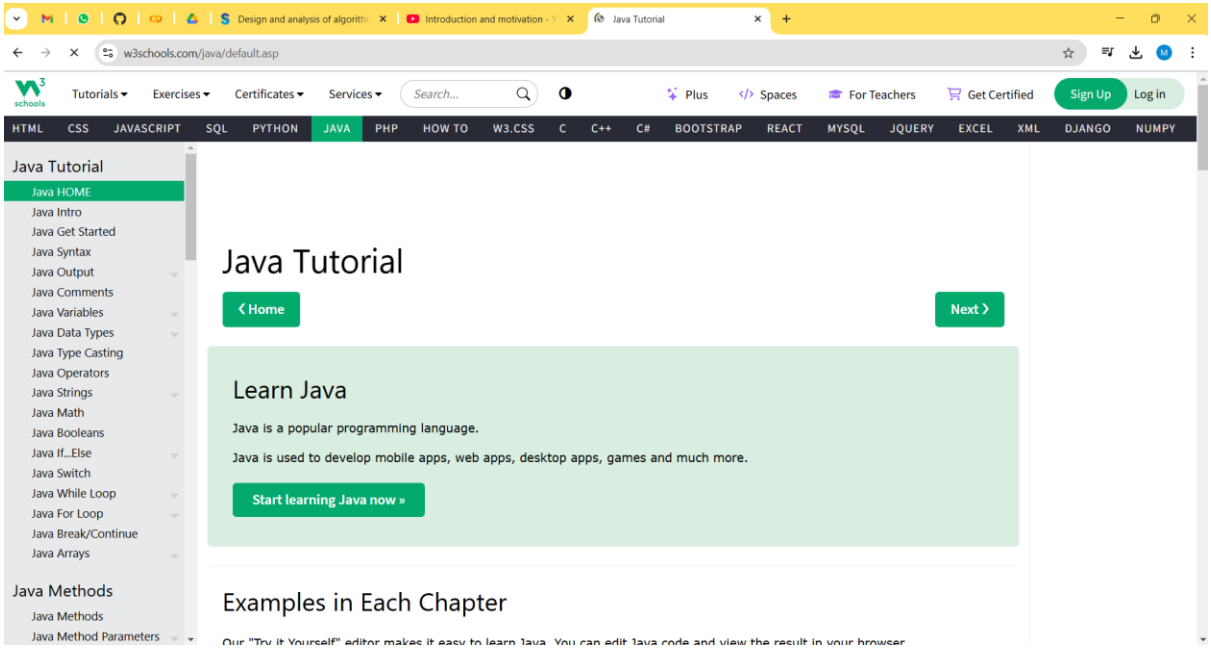
RTGS Details:

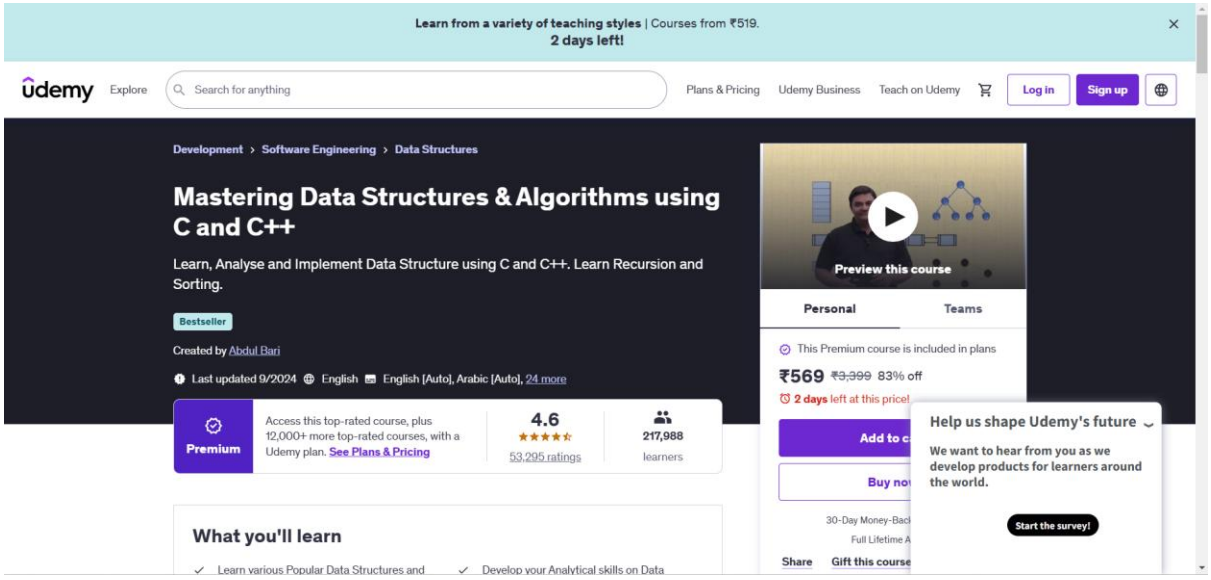
Bank Name: State Bank of India
Bank branch address: The Capital, 16th Floor BKC,Bandra East, Dist Mumbai 400051
Account no: 40824110781 RTGS/IFSC Code: SBIN0016376
Swift code: SBININBB796 MICR Code: NA

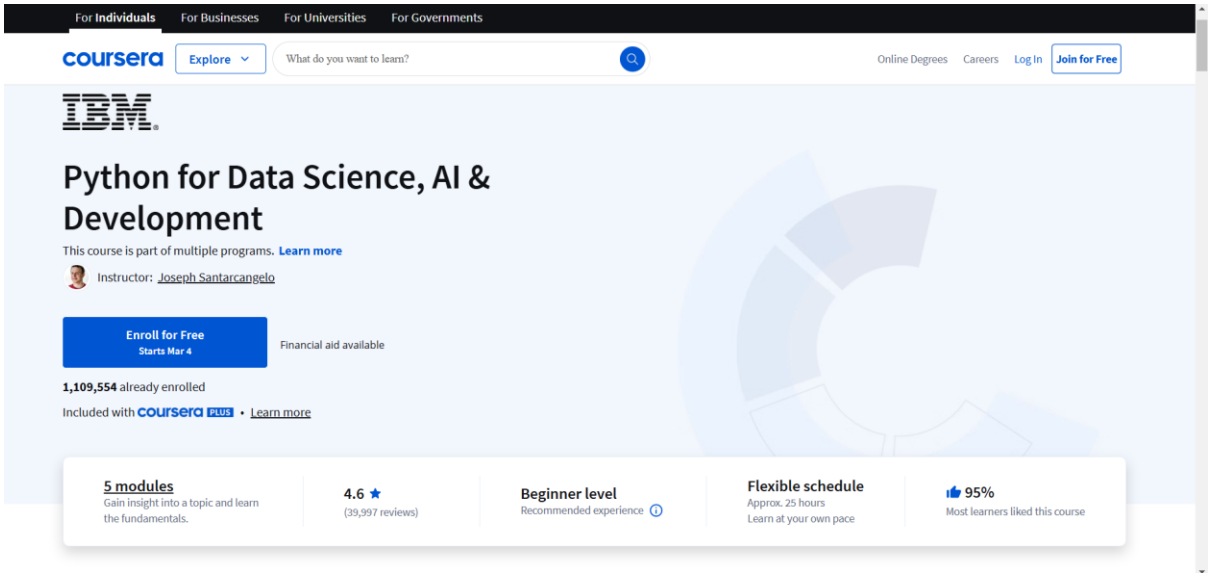
GSTIN: 34AAACB2100P1Z2

Knowledgewall

Knowledge Wall






For Individuals
For Businesses
For Universities
For Governments
coursera
Explore
What do you want to learn?
Online Degrees
Careers
Log In
Join for Free
IBM
Python for Data Science, AI & Development
This course is part of multiple programs. Learn more
Instructor: Joseph Santarcangelo
Enroll for Free
Starts Mar 4
Financial aid available
1,109,554 already enrolled
Included with coursera PLUS • Learn more
5 modules
Gain insight into a topic and learn the fundamentals.
4.6 ★
(39,997 reviews)
Beginner level
Recommended experience
Flexible schedule
Approx. 25 hours
Learn at your own pace
95%
Most learners liked this course

LCD Projector


GPS Map Camera
Pangur, Puducherry, India
Wp33+8ff, Pangur, Puducherry 605110, India
Lat 11.902238° Long 79.70303°
04/03/2025 11:19 AM GMT +05:30
Google


GPS Map Camera
Pangur, Puducherry, India
Wp33+8ff, Pangur, Puducherry 605110, India
Lat 11.902117° Long 79.702864°
04/03/2025 11:18 AM GMT +05:30
Google