

OJT- Practicals

C AND C++

(1)

Aim- Write a C programs to print the address of a variable using a pointer.

Code:

```
#include<stdio.h>
int main(){
int num=10;
int *p;
p=&num;
printf ("the value of number is=%d\n",num);
printf ("the address of number is=%x\n",p);
printf ("the address is=%d\n",*p);
return 0;
}
```

Output:

```
the value of number is=10
the address of number is=61ff18
the address is=10
```

(2)

Aim: Write a C program to create a Calculator using a pointer.

Code:

```
#include <stdio.h>

int main() {
    int num1, num2, result;
    char operator;
    int* pResult = &result;

    printf("Enter first number: ");
    scanf("%d", &num1);

    printf("Enter operator (+, -, *, /): ");
    scanf(" %c", &operator);

    printf("Enter second number: ");
    scanf("%d", &num2);

    switch(operator) {
        case '+':
            *pResult = num1 + num2;
            break;
        case '-':
            *pResult = num1 - num2;
            break;
        case '*':
            *pResult = num1 * num2;
            break;
        case '/':
            *pResult = num1 / num2;
            break;
        default:
            printf("Invalid operator!");
            return 1;
    }
}
```

```
    printf("%d %c %d = %d\n", num1, operator, num2,
*pResult);

    return 0;
}
```

Output:

Enter first number: 4

Enter operator (+, -, *, /): +

Enter second number: 6

4 + 6 = 10

(3)

Aim: Write a C program to swap the two values using call by value and call by reference

Code:

```
#include<stdio.h>
void call_by_value(int x,int y){
    int temp;
    temp=x;
    x=y;
    y=temp;
    printf("After Swaping The Value By Value val1=%d
val2=%d\n",x,y);
}
void call_by_refrence(int *x,int *y)
```

```

{
    int temp;
    temp=*x;
    *x=*y;
    *y=temp;
    printf("After Swaping The Value By Refrence
val1=%d val2=%d",*x,*y);
}
void main()
{
    int val1,val2;
    printf("Enter the val 1=");
    scanf("%d",&val1);
    printf("Enter The val 2=");
    scanf("%d",&val2);
    call_by_value(val1,val2);
    call_by_refrence(&val1,&val2);
}

```

Output:

Enter the val 1=4

Enter The val 2=5

After Swaping The Value By Value val1=5 val2=4

After Swaping The Value By Refrence val1=5 val2=4

(4)

Aim: Define a structure type struct personal that would contain person name, Date of birth and age using this Structure to read this information of 4 people and display the same.

Code:

```
#include<stdio.h>
```

```
#include<string.h>
struct employe
{
    char name[50];
    int dob;
    int age;
}e1[2];
void get(struct employe e[]);
void main()
{
    int i;
    for(i=0;i<4;i++)
    {
        printf("\n Enter The Name=");
        scanf("%s",&e1[i].name);
        printf("\n Enter The DOB=");
        scanf("%d",&e1[i].dob);
        printf("\n Enter The Age=");
        scanf("%d",&e1[i].age);
    }
    get(e1);
}

void get(struct employe e1[])
{
    int i;
    for(i=0;i<4;i++)
    {
        printf("Employee name=%s\n",e1[i].name);
        printf("Employee DOB=%d\n",e1[i].dob);
    }
}
```

```
printf("Employee Age=%d\n",e1[i].age);  
}  
}
```

(5)

Aim: Write a C program to calculate the sum of n numbers entered by the user using dynamic memory allocation.

Code:

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main () {  
    int n, i, *ptr, sum = 0;  
  
    printf("Enter the number of elements: ");  
    scanf("%d", &n);  
  
    ptr = (int*) malloc(n * sizeof(int));  
  
    if (ptr == NULL) {  
        printf("Memory allocation failed!\n");  
        return 1;  
    }  
  
    printf("Enter the elements:\n");  
    for (i = 0; i < n; i++) {  
        scanf("%d", ptr + i);  
        sum += *(ptr + i);  
    }  
}
```

```
printf("Sum = %d\n", sum);

// Free dynamically allocated memory
free(ptr);

return 0;
}
```

Output:

Enter the number of elements: 5

Enter the elements:

1

2

3

4

5

Sum = 15

(6)

Aim: A file named “New” contains a series of integer numbers. Write

a c program to read all numbers from a file and then copy all odd numbers into a file named “odd” and write all even numbers into a file named “even”. Then display the values of files odd and even on the screen.

Code:

```
#include <stdio.h>

int main() {
    FILE *fp, *fpo, *fpe;
    int num;

    fp = fopen("New.txt", "r");
    if(fp == NULL) {
        printf("Error opening file!\n");
        return 1;
    }

    fpo = fopen("odd.txt", "w");
    if(fpo == NULL) {
        printf("Error creating odd file!\n");
        return 1;
    }

    fpe = fopen("even.txt", "w");
    if(fpe == NULL) {
        printf("Error creating even file!\n");
        return 1;
    }

    while(fscanf(fp, "%d", &num) != EOF) {
        if(num % 2 == 0) {
            fprintf(fpe, "%d\n", num);
        }
    }
}
```



```
        } else {
            fprintf(fpo, "%d\n", num);
        }
    }

    fclose(fp);
    fclose(fpo);
    fclose(fpe);

    // Display contents of odd file
    printf("Odd numbers:\n");
    fpo = fopen("odd.txt", "r");
    if(fpo == NULL) {
        printf("Error opening odd file!\n");
        return 1;
    }
    while(fscanf(fpo, "%d", &num) != EOF) {
        printf("%d\n", num);
    }
    fclose(fpo);

    // Display contents of even file
    printf("\nEven numbers:\n");
    fpe = fopen("even.txt", "r");
    if(fpe == NULL) {
        printf("Error opening even file!\n");
        return 1;
    }
    while(fscanf(fpe, "%d", &num) != EOF) {
        printf("%d\n", num);
    }
    fclose(fpe);

    return 0;}
```

Output:

Odd numbers:

5

7

3

9

1

Even numbers:

4

8

6

2

10

(7)

Aim: Write a C++ program to Check if the number is prime or not using a function.

Code:

```
#include <iostream>
using namespace std;
void Prime(int num) {
```

```

    if (num <= 1) {
        cout << num << " is not a prime number" <<
endl;
        return;
    }

    for (int i = 2; i <= num/2; i++) {
        if (num % i == 0) {
            cout << num << " is not a prime number"
<< endl;
            return;
        }
    }
    cout << num << " is a prime number" << endl;
}

int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;
    Prime(num);
    return 0;
}

```

Output:

Enter a number: 0

0 is not a prime number

(8)

Aim: Write a C++ program that prompts the user to enter a letter and check whether a letter is a vowel or constant.

Code:

```
#include<iostream>
using namespace std;
void Vowel(){
    char c;
    cout<<"Enter The Character=";
    cin>>c;
    if(c=='a' || c=='e' || c=='i' || c=='o' || c=='u' ||
c=='A' || c=='E' || c=='I' || c=='O' || c=='U')
    {
        cout<<"Character Is Vowel";
    }
    else
    {
        cout<<"The Character Is Consonant";
    }
}

int main()
{
    Vowel();
    return 0;
}
```

Output:

Enter The Character=a

Character Is Vowel

Enter The Character=q

The Character Is Consonant

(9)

Aim: Write a C++ program to demonstrate the concept of constructor and destructor.

Code:

```
#include <iostream>
using namespace std;

class demo {
public:
    demo() {
        cout << "Constructor called" << endl;
    }
    ~demo() {
        cout << "Destructor called" << endl;
    }
};

int main() {
    demo obj;
    return 0;
}
```

Output:

Constructor called

Destructor called

(10)

Aim: Create a class student that stores roll_no, name. Create a class test that stores marks obtained in five subjects. Class result derived from student and test contains the total marks and percentage obtained in test. Input and display information of a student.

Code:

```
#include <iostream>
#include <string>
using namespace std;

class Student {
public:
    int roll_no;
    string name;
};

class Test {
public:
    int marks[5];
};

class Result : public Student, public Test {
public:
    int total_marks;
    float percentage;

    void calculateResult() {
        total_marks = 0;
        for(int i = 0; i < 5; i++) {
            total_marks += marks[i];
        }
    }
};
```

```

        percentage = (float) total_marks / 5;
    }

    void displayResult() {
        cout << "Roll Number: " << roll_no << endl;
        cout << "Name: " << name << endl;
        cout << "Total Marks: " << total_marks << endl;
        cout << "Percentage: " << percentage << "%" <<
endl;
    }
};

int main() {
    Result student_result;

    cout << "Enter Roll Number: ";
    cin >> student_result.roll_no;

    cout << "Enter Name: ";
    cin.ignore(); // ignore the newline character left
in the stream by previous input
    getline(cin, student_result.name);

    cout << "Enter Marks Obtained in 5 Subjects: " <<
endl;
    for(int i = 0; i < 5; i++) {
        cout << "Subject " << i+1 << ": ";
        cin >> student_result.marks[i];
    }

    student_result.calculateResult();
    student_result.displayResult();

    return 0;}

```

Output:

Enter Roll Number: 14

Enter Name: Jatin

Enter Marks Obtained in 5 Subjects:

Subject 1: 12

Subject 2: 34

Subject 3: 23

Subject 4: 43

Subject 5: 23

Roll Number: 14

Name: Jatin

Total Marks: 135

Percentage: 27%

(11)

Aim: Write a C++ program to overload binary + operator.

Code:

```
#include <iostream>
using namespace std;

class Number
{
private:
```



```

    int value;

    public:
    Number(int v = 0) : value(v) {}

    Number operator+(const Number& n) {
        return Number(value + n.value);
    }

    void display() {
        cout << "Value: " << value << endl;
    }
};

int main() {
    Number n1(5), n2(10), n3;
    n3 = n1 + n2;
    n3.display();
    return 0;
}

```

Output:

Value: 15

(12)

Aim: Create a base class called 'SHAPE' having two data members of type double, member function get_data() to initialize base class data members, pure virtual member function display_area() to compute and display the area of the geometrical object. Derive two specific classes 'TRIANGLE' and 'RECTANGLE' from the base class.

Using these three classes design a program that will accept dimension of a triangle / rectangle interactively and display the area.

Code:

```
#include <iostream>
#include <cmath>
using namespace std;

class SHAPE {
protected:
    double length;
    double width;

public:
    void get_data() {
        cout << "Enter length: ";
        cin >> length;
        cout << "Enter width: ";
        cin >> width;
    }

    virtual void display_area() = 0;
};

class TRIANGLE : public SHAPE {
public:
    void display_area() {
        double area = 0.5 * length * width;
        cout << "Area of triangle: " << area << endl;
    }
};

class RECTANGLE : public SHAPE {
public:
```

```

        void display_area() {
            double area = length * width;
            cout << "Area of rectangle: " << area <<
endl;
        }
};

int main() {
    SHAPE *s;
    TRIANGLE t;
    RECTANGLE r;
    int choice = 0;

    while (choice != 1 && choice != 2) {
        cout << "Enter shape (1 = Triangle, 2 =
Rectangle): ";
        cin >> choice;
    }

    if (choice == 1) {
        s = &t;
    } else {
        s = &r;
    }
    s->get_data();
    s->display_area();

    return 0;
}

```

Output:

Enter shape (1 = Triangle, 2 = Rectangle): 1

Enter length: 12

DBMS

(13)

AIM :- To study DDL-create and DML-insert commands.

DDL (Data Definition Language) and DML (Data Manipulation Language) are two different types of

SQL (Structured Query Language) commands used to manage and manipulate databases.

DDL commands are used to define and manipulate the structure of the database, including creating

tables, altering tables, adding columns, and deleting tables.

DML commands, on the other hand, are used to manipulate the data within the database, including

inserting, updating, and deleting data in tables.

Let's start by looking at the syntax and usage of DDL-create and DML-insert commands:

DDL-Create Command:

The create command is used to create a new table in the database. The basic syntax of the create

command is as follows:

CODE :-

```
CREATE TABLE table_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
    ....  
);
```

For example, to create a table named "customers" with columns for "id", "name", "email", and "phone", the following command would be used:

CODE :-

```
CREATE TABLE customers (  
    id INT PRIMARY KEY,  
    name VARCHAR(50),  
    email VARCHAR(50),  
    phone VARCHAR(20)  
);
```

DML-Insert Command:

The insert command is used to insert data into a table in the database. The basic syntax of the insert command is as follows:

CODE :-

```
INSERT INTO table_name (column1, column2, column3, ...)
```

```
VALUES (value1, value2, value3, ...);
```

For example, to insert a new row of data into the "customers" table created in the previous

example, the following command would be used:

CODE :-

```
INSERT INTO customers (id, name, email, phone)
```

```
VALUES (1, 'John Doe', 'john@example.com', '555-1234');
```

This command would insert a new row into the "customers" table with the values specified for the

"id", "name", "email", and "phone" columns.

I hope this helps you understand the basics of DDL-create and DML-insert commands in SQL. If you

have any further questions, feel free to ask!

(14)

Aim:Create following Table

(A). Job (job_id, job_title, min_sal, max_sal)

COLUMN NAME	DATA TYPE
-------------	-----------

job_id	Varchar(15)
job_title	Varchar(30)
min_sal	Int
max_sal	Int

Code:

```
CREATE TABLE Job (
  job_id VARCHAR(15) PRIMARY KEY,
  job_title VARCHAR(30),
  min_sal INT,
  max_sal INT
);
```

(B). Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no)

COLUMN NAME	DATA TYPE
emp_no	Int
emp_name	Varchar(30)
emp_sal	decimal(8,2)

emp_comm	decimal(6,1)
dept_no	Int

Code:

```
CREATE TABLE Employee (
    emp_no INT PRIMARY KEY,
    emp_name VARCHAR(30),
    emp_sal DECIMAL(8,2),
    emp_comm DECIMAL(6,1),
    dept_no INT
);
```

(C).deposit(a_no,cname,bname,amount,a_date)

COLUMN NAME	DATA TYPE
a_no	Int,identity
C_name	Varchar(50)
B_name	Varchar(30)
amount	Decimal(4,2)

a_date	Date
--------	------

Code:

```
CREATE TABLE deposit (
a_no INT IDENTITY PRIMARY KEY,
cname VARCHAR(50),
bname VARCHAR(30),
amount DECIMAL(4,2),
a_date DATE
```

(D). borrow(loanno,cname,bname,amount)

COLUMN NAME	DATA TYPE
loanno	Int
cname	Varchar(25)
bname	Varchar(20)
amount	Decimal(6,2)

Code:

```
CREATE TABLE borrow (
loanno INT PRIMARY KEY,
```

```
cname VARCHAR(25),  
bname VARCHAR(20),  
amount DECIMAL(6,2)  
);
```

(16)

Aim:Create tables and insert sample data in tables.

Write SQL queries to insert following data into tables

Code:

```
CREATE TABLE Employee (  
    emp_no INT PRIMARY KEY,  
    emp_name VARCHAR(30) NOT NULL,  
    emp_sal DECIMAL(8,2) NOT NULL,  
    emp_comm DECIMAL(6,1),  
    dept_no INT NOT NULL  
);  
  
INSERT INTO Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no) VALUES  
  
    (101, 'Smith', 800.00, 20, 0),  
  
    (102, 'Snehal', 1600.00, 300, 25),  
  
    (103, 'Adama', 1100.00, 0, 20),  
  
    (104, 'Aman', 3000.00, 15, 0),  
  
    (105, 'Anita', 5000.00, 50000, 10),  
  
    (106, 'Sneha', 2450.00, 24500, 10),
```

(107, 'Anamika', 2975.00, 30, 0);

(b).

AIM :- Create tables and insert sample data in tables. Write SQL queries to insert following data into tables Insert following values in the table Job

job_id	job_name	min_sal	max_sal
IT_PROG	Programmer	4000	10000
MK_MGR	Marketing manager	9000	15000
FI_MGR	Finance manager	8200	12000
FI_ACC	Account	4200	9000
LEC	Lecturer	6000	17000
COMP_OP	Computer Operator	1500	3000

CODE:

```
CREATE TABLE Job (
```

```
  job_id VARCHAR(15) PRIMARY KEY,
```

```
  job_name VARCHAR(30) NOT NULL,
```

```
  min_sal DECIMAL(10,2) NOT NULL,
```

```
  max_sal DECIMAL(10,2) NOT NULL
```

```
);
```

```
INSERT INTO Job (job_id, job_name, min_sal, max_sal) VALUES
```

```
('IT_PROG', 'Programmer', 4000.00, 10000.00),
```

```
('MK_MGR', 'Marketing manager', 9000.00, 15000.00),
```

```
('FI_MGR', 'Finance manager', 8200.00, 12000.00),
```

```
('FI_ACC', 'Account', 4200.00, 9000.00),
```

```
('LEC', 'Lecturer', 6000.00, 17000.00),
```

('COMP_OP', 'Computer Operator', 1500.00, 3000.00);

(c).

Aim:

A_no	cname	Bname	Amount	date
101	Anil	andheri	7000	01-jan-06
102	sunil	virar	5000	15-jul-06
103	jay	villeparle	6500	12-mar-06
104	vijay	andheri	8000	17-sep-06
105	keyur	dadar	7500	19-nov-06
106	mayur	borivali	5500	21-dec-06

Code:

```
CREATE TABLE deposit (
```

```
  a_no INT IDENTITY PRIMARY KEY,
```

```
  cname VARCHAR(50) NOT NULL,
```

```
  bname VARCHAR(30) NOT NULL,
```

```
  amount DECIMAL(8,2) NOT NULL,
```

```
  a_date DATE NOT NULL
```

```
);
```

```
INSERT INTO deposit (cname, bname, amount, a_date) VALUES
```

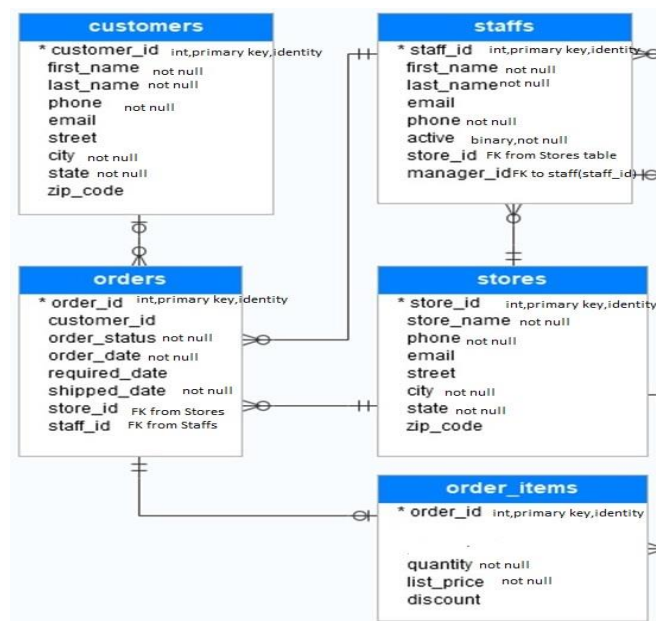
```
  ('Anil', 'andheri', 7000.00, '2006-01-01'),
```

```
  ('sunil', 'virar', 5000.00, '2006-07-15'),
```

('jay', 'villeparle', 6500.00, '2006-03-12'),
 ('vijay', 'andheri', 8000.00, '2006-09-17'),
 ('keyur', 'dadar', 7500.00, '2006-11-19'),
 ('mayur', 'borivali', 5500.00, '2006-12-21');

(16)

Aim: Write the SQL queries to provide constraints on given tables. Create A Database Sales and Write SQL Queries to create following tables with all constrains mentioned in image.



Code:

```

CREATE TABLE customers (
  customer_id INT PRIMARY KEY,
  first_name VARCHAR(50) NOT NULL,

```

```
last_name VARCHAR(50) NOT NULL,  
phone VARCHAR(20),  
email VARCHAR(100),  
street VARCHAR(100),  
city VARCHAR(50) NOT NULL,  
state VARCHAR(50) NOT NULL,  
zip_code VARCHAR(20)  
);
```

```
CREATE TABLE staff (  
    staff_id INT PRIMARY KEY,  
    first_name VARCHAR(50) NOT NULL,  
    last_name VARCHAR(50) NOT NULL,  
    email VARCHAR(100),  
    phone VARCHAR(20) NOT NULL,  
    active BOOLEAN NOT NULL,  
    store_id INT,  
    FOREIGN KEY (store_id) REFERENCES stores(store_id)  
);
```

```
CREATE TABLE stores (  
    store_id INT PRIMARY KEY,  
    store_name VARCHAR(50) NOT NULL,  
    phone VARCHAR(20) NOT NULL,  
    email VARCHAR(100),  
    street VARCHAR(100),
```

```
city VARCHAR(50) NOT NULL,  
state VARCHAR(50) NOT NULL,  
zip_code VARCHAR(20),  
manager_id INT,  
FOREIGN KEY (manager_id) REFERENCES staff(staff_id)  
);
```

```
CREATE TABLE orders (  
order_id INT PRIMARY KEY,  
order_date DATE NOT NULL,  
required_date DATE,  
shipped_date DATE NOT NULL,  
order_status VARCHAR(20) NOT NULL,  
customer_id INT,  
staff_id INT,  
store_id INT,  
FOREIGN KEY (customer_id) REFERENCES customers(customer_id),  
FOREIGN KEY (staff_id) REFERENCES staff(staff_id),  
FOREIGN KEY (store_id) REFERENCES stores(store_id)  
);
```

```
CREATE TABLE order_items (  
order_id INT,  
item_id INT PRIMARY KEY,  
quantity INT NOT NULL,  
list_price DECIMAL(10, 2) NOT NULL,
```

```
discount DECIMAL(5, 2),  
FOREIGN KEY (order_id) REFERENCES orders(order_id)  
);
```

(17)

Aim: Write the SQL queries to perform various aggregate functions on table data.

1. List total deposit from deposit.
2. List total amount from andheri branch
3. Count total number of customers
4. Count total number of customer's cities.
5. Update the value dept_no to 10 where second character of emp. name is 'm'.
6. Update the value of employee name whose employee number is 103.
7. Write a query to display the current date. Label the column Date
8. For each employee, display the employee number, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary
9. Modify your previous query to add a column that subtracts the old salary from the new salary. Label the column Increment.

Code:

AIM :- Write the SQL queries to perform various aggregate functions on table data.

1. List total deposit from deposit.

CODE :- `SELECT SUM(amount) AS total_deposit FROM deposit;`

2. List total amount from andheri branch

CODE :- SELECT SUM(amount) AS total_amount FROM deposit WHERE bname = 'andheri';

3. Count total number of customers

CODE :- SELECT COUNT(DISTINCT cname) AS total_customers FROM deposit;

4. Count total number of customer's cities

CODE :- SELECT COUNT(DISTINCT bname) AS total_cities FROM deposit;

5. Update the value dept_no to 10 where second character of emp. name is 'm'.

CODE :- UPDATE Employee SET dept_no = 10 WHERE emp_name LIKE '_m%';

6. Update the value of employee name whose employee number is 103.

CODE :- UPDATE Employee SET emp_name = 'Adam' WHERE emp_no = 103;

7. Write a query to display the current date. Label the column Date

CODE :- SELECT GETDATE() AS Date;

8. For each employee, display the employee number, salary, and salary in

creased by 15% and expressed as a whole number. Label the column New Salary

CODE :- SELECT emp_no, emp_sal, ROUND(emp_sal*1.15,0) AS "New Salary" FROM Employee;

9. Modify your previous query to add a column that subtracts the old salary from the new salary. Label the column Increment.

CODE :- SELECT emp_no, emp_sal, ROUND(emp_sal*1.15,0) AS "New Salary",
ROUND(emp_sal*0.15,0) AS Increment FROM Employee;

(18)

1. Retrieve all data from employee, jobs and deposit.

CODE :- SELECT * FROM employee;

SELECT * FROM jobs;

SELECT * FROM deposit;

2. Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.

CODE :- SELECT a_no, amount FROM deposit

WHERE a_date BETWEEN '2006-01-01' AND '2006-07-25';

3. Display all jobs with minimum salary is greater than 4000.

CODE:- SELECT * FROM jobs

WHERE min_sal > 4000;

4. Display name and salary of employee whose department no is 20. Give alias name to name of employee.

CODE :- SELECT emp_no, emp_name AS employee_name, emp_sal, dept_no
FROM

employee

WHERE dept_no = 20;

5. Display employee no,name and department details of those employee whose department lies in(10,20)

CODE :- SELECT emp_no, emp_name, dept_no FROM employee

WHERE dept_no IN (10, 20);

6. Display all employee whose name start with 'A' and third character is 'a'.

CODE :- SELECT * FROM employee

WHERE emp_name LIKE 'A_a%';

7. Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.

CODE :-

```
SELECT emp_name, emp_no, emp_sal FROM employee  
WHERE emp_name LIKE 'Ani_____';
```

8. Display the non-null values of employees and also employee name second characters should be 'n' and string should be 5 character long.

CODE :-

```
SELECT * FROM employee  
WHERE emp_name LIKE '_n%' AND LENGTH(emp_name) = 5 AND emp_name IS  
NOT NULL;
```

9. Display the null values of employee and also employee name's third character should be 'a'.

CODE :-

```
SELECT * FROM employee  
WHERE emp_name LIKE '___a%' AND emp_name IS NULL;
```

HTML || CSS || JS

(18)

Aim : Make a Resume using the HTML tags without CSS.

Code :

```
<!doctype html>
<html>
<head>
<title>resume</title>
</head>
<body>

<fieldset><center><h1><u><b>RESUME</b></u></h1></center>
  <span>Photo:</span>
  <img src=""/>
<br>
  <span>First Name:</span>
  <input type="text area"/>
<br>
  <span>Second Name:</span>
  <input type="text area"/>
<br>
  <span>DOB:</span>
  <input type="date"/>
<br>
  <span>E-Mail</span>
  <input type="text area"/>
<br>
  <span>Gender:</span>
  <span>Male</span>
  <input type="radio" id="g1" name="gender"/>
  <span>Female</span>
  <input type="radio" id="g2" name="gender"/>
<br>
  <span>Address:</span>
  <textarea plceholder="Enter The Address"></textarea>
<br>
  <span>Hobby:</span>
```

```

☐Reading
☐Dancing
☐Gaming
☐Cricket
<hr>
<center><h3><u><b>SKILLS</b></u></h3></center>
<ol type="square">
<li>Technical Skills</li>
<li>Soft Skills</li>
<li>Aptitude Skills</li>
<li>Communication Skills</li>
</ol>
<hr>
<center><h3><u><b>QUAILIFICATIONS</b></u></h3></center>
<table border=1 align="left">
  <thead>
    <tr>
      <th>s.no</th>
      <th>Class</th>
      <th>Grade</th>
      <th>Board</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>B.Tech</td>
      <td>Running</td>
      <td>SOU</td>
    </tr>
    <tr>
      <td>2</td>
      <td>12<sup>th</sup></td>
      <td>B</td>
      <td>GSHEB</td>
    </tr>
    <tr>
      <td>3</td>
      <td>10<sup>th</sup></td>
      <td>A</td>
      <td>GSHEB</td>
    </tr>
  </tbody>
</table>

```

```

    </table>
</fieldset>

</body>
</html>

```

Output:

RESUME

Photo:
 First Name:
 Second Name:
 DOB:
 E-Mail:
 Gender: Male ☐ Female ☐
 Address:
 Hobby: ☐ Reading ☐ Dancing ☐ Gaming ☐ Cricket

SKILLS

1. Technical Skills
 2. Soft Skills
 3. Aptitude Skills
 4. Communication Skills

QUAILIFICATIONS

s.no	Class	Grade	Board
1	B.Tech	Running	SOU
2	12 th	B	GSHEB
3	10 th	A	GSHEB

(19)

Aim : Create an HTML webpage that shows Poster Presentation using all Table Properties.

Code :

```

<!doctype html>
<html>
  <head>
    <title> Table </title>
  </head>
<body>

```

```
<table border=1 align="center">
  <thead>
    <tr>
      <th>s.no</th>
      <th>first name</th>
      <th>last name</th>
      <th>email</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>jatin </td>
      <td rowspan="2">sharma</td>
      <td>jatin6969</td>
    </tr>

    <tr>
      <td>2</td>
      <td>ajay </td>
      <td>ajay6969</td>
    </tr>

    <tr>
      <td>3</td>
      <td>adi </td>
      <td>gond</td>
```

```

        <td>adi6969</td>
        </tr>
        <td>4</td>
        <td colspan="3">ashish kushwah ashish6969
</td>
        </tr>
        </table>

</body>
</html>

```

Output :

s.no	first name	last name	email
1	jatin	sharma	jatin6969
2	ajay		ajay6969
3	adi	gond	adi6969
4	ashish kushwah ashish6969		

(20)

Aim : Create an HTML page table and form.

Code :

```
<!doctype html>
<html>
  <head>
    <title>FORM TAGS</title>
  </head>
<body>
<form>
  <span>First Name:</span>
    <input type="text" autofocus placeholder="Enter The First Name"/>
  <br><span>Last Name:</span>
    <input type="text" autofocus placeholder="Enter The Last Name"/>
  <br>
  <span>Gender:</span>
  <span>Male</span>
    <input type="radio" id="g1" name="gender">
  <span>Female</span>
    <input type="radio" id="g1" name="gender">
  <br>
  <span>Hobby:</span>
    <input type="checkbox" />Cricket
    <input type="checkbox" />Badminton
    <input type="checkbox" />Football
    <input type="checkbox" />Kabadi
  <br><span>Address:</span>
    <textarea plceholder="Enter The Address"></textarea>

  <br><span>City:</span>
  <select>
    <option> ahemdabad</option>
    <option>surat</option>
    <option>rajkot</option>
  </select>
  <br>
  <span>File:</span>
    <input type="file" multi accept=".jpg,.png,.pdf,.gif ">
  <br>
  <span>Password:</span>
    <input type="password"/>
  <br>
  <span>Date:</span>
    <input type="date">
  <br>
```

```

<span>Time:</span>
  <input type="time">
<br>
<span>Color:</span>
  <input type="color">
<br>
<span>Range:</span>
  <input type="range"/>
<fieldset>
  <legend>sign in</legend>
  <span>username</span><br>
    <input type="text"/><br>
  <span>Password</span><br>
    <input type="password"/><br>
  <input type="button" value="sign in"/>
</fieldset>
</form>
</body>
</html>

```

Output :

First Name:

Last Name:

Gender: Male ☐ Female ☐

Hobby: ☐ Cricket ☐ Badminton ☐ Football ☐ Kabadi

Address:

City:

File: No file chosen

Password:

Date:

Time:

Color:

Range:

sign in

username

 Password

 sign in

(21)

Aim : Create Registration form and do proper validation with HTML 5 inbuilt functionality. (Don't use JavaScript).

Code :

```
<!DOCTYPE html>
<head>
  <title>Registration Form</title>
  <style>
    #form
    {
        width: 320px;
        background-color: antiquewhite;
        color: #000000;
        margin: 50px auto;
        padding: 20px;
        text-align: center;
        border-radius: 10px;
        border: 2px solid #000000;
    }
    h1{
        text-align: center;
        margin-bottom: 20px;
    }
  </style>
  <script>
    function validation()
    {
        var name=document.forms.regform.Name.value;
        var address=document.forms.regform.address.value;
        var email=document.forms.regform.email.value;
        var mobile=document.forms.regform.mobile.value;
        var password=document.forms.regform.password.value;
        var subject=document.forms.regform.subject.value;
        alert(name + " " + address + " " + email + " " + mobile + " " + password + "
"+subject);

        if(name==" " || name=="null")
        {
            alert("Must Enter The Name");
        }
        if(address==" " || address=="null")
        {
            alert("Must Enter The Address");
        }
        if(email==" " || email=="null")
        {
            alert("Must Enter The E-mail");
        }
        if(mobile==" " || mobile=="null")
        {
            alert("Must Enter The Mobile");
        }
    }
  </script>

```

```

        if(password==" " || password=="null")
        {
            alert("Must Enter The Password");
        }
        if(subject==" " || subject=="null")
        {
            alert("Must Enter The Subject");
        }
    }
</script>
</head>
<body>
    <div id="form">
        <h1>Registration Form</h1>
        <form name="regform" onsubmit="return validation()" method="post">
            Name:<input type="text" name="Name"/><br>
            Address:<input type="text" name="address"/><br>
            E-mail:<input type="text" name="email"/><br>
            Mobile:<input type="text" name="mobile"/><br>
            Password:<input type="text" name="password" placeholder="password"/><br>
            <label for="subject">Select a course:</label>
            <select id="subject" name="subject" >
                <option value="">Choose-Course</option>
                <option value="math">Math</option>
                <option value="english">English</option>
                <option value="history">History</option>
                <option value="science">Science</option>
            </select>

            <br>
            <input type="submit" value="send" name="submit"/><br>
        </form>
    </div>
</body>
</html>

```

Output :

Registration Form

Name:

Address:

E-mail:

Mobile:

Password:

Select a course:

(22)

Aim : Make a Resume using the HTML tags with CSS.

Code :

```
<!doctype html>
<html>
<head>
<title>resume</title>
<style>
    body {
        font-family: Arial, sans-serif;
        background-color: #f7f7f7;
    }

    fieldset {
        margin: 50px auto;
        padding: 20px;
        background-color: #fff;
        width: 500px;
        box-shadow: 0 0 10px rgba(0,0,0,0.1);
        border-radius: 10px;
        border: none;
    }

    h1 {
        font-size: 36px;
        margin-bottom: 20px;
    }

    h3 {
```

```
    font-size: 24px;
    margin-top: 30px;
    margin-bottom: 20px;
}

input[type="text"], input[type="email"], input[type="password"],
select, textarea {
    width: 100%;
    padding: 10px;
    border-radius: 5px;
    border: none;
    background-color: #f3f3f3;
    margin-bottom: 20px;
    font-size: 16px;
}

input[type="radio"], input[type="checkbox"] {
    margin-right: 10px;
}

textarea {
    height: 100px;
}

ol {
    margin-left: 20px;
}

table {
    width: 100%;
    border-collapse: collapse;
```

```
        margin-top: 30px;
    }

    th, td {
        padding: 10px;
        text-align: left;
        border-bottom: 1px solid #ddd;
    }

    th {
        background-color: #f3f3f3;
    }

    tr:hover {
        background-color: #f5f5f5;
    }

    button[type="submit"] {
        background-color: #4CAF50;
        color: #fff;
        padding: 10px 20px;
        border: none;
        border-radius: 5px;
        cursor: pointer;
        font-size: 16px;
        margin-top: 20px;
    }

    button[type="submit"]:hover {
        background-color: #3e8e41;
    }
}
```

```
</style>
</head>
<body>

    <div>
        <fieldset><center><h1><u><b>RESUME</b></u></h1></center>
        <span>Photo:</span>
        
    <br>
        <span>First Name:</span>
        <input type="text area"/>
    <br>
        <span>Second Name:</span>
        <input type="text area"/>
    <br>
        <span>DOB:</span>
        <input type="date"/>
    <br>
        <span>E-Mail</span>
        <input type="text area"/>
    <br>
        <span>Gender:</span>
        <span>Male</span>
        <input type="radio" id="g1" name="gender"/>
        <span>Female</span>
        <input type="radio" id="g2" name="gender"/>
    <br>
        <span>Address:</span>
        <textarea plceholder="Enter The Address"></textarea>
    <br>
        <span>Hobby:</span>
```



```

<input type="checkbox"/>Reading
<input type="checkbox"/>Dancing
<input type="checkbox"/>Gaming
<input type="checkbox"/>Cricket
<hr>
<center><h3><u><b>SKILLS</b></u></h3></center>
<ol type="square">
<li>Technical Skills</li>
<li>Soft Skills</li>
<li>Aptitude Skills</li>
<li>Comunication Skills</li>
</ol>
<hr>
<center><h3><u><b>QUAILIFICATIONS</b></u></h3></center>
<table border=1 align="left">
  <thead>
    <tr>
      <th>s.no</th>
      <th>Class</th>
      <th>Grade</th>
      <th>Board</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>B.Tech</td>
      <td>Running</td>
      <td>SOU</td>
    </tr>
  </tbody>
</table>

```

```

                <td>2</td>
            <td>12<sup>th</sup> </td>
            <td>B</td>
            <td>GSHEB</td>
        </tr>

    <tr>

        <td>3</td>
        <td>10<sup>th</sup> </td>
        <td>A</td>
        <td>GSHEB</td>
    </tr>

</tbody>
</table>
</fieldset>




</div>

</body>
</html>

```

Output :

RESUME

Photo: 
 First Name:
 Second Name:
 DOB: 
 E-Mail:
 Gender: Male ☐ Female ☐
 Address: 

Hobby: ☐ Reading ☐ Dancing ☐ Gaming ☐ Cricket

SKILLS

1. Technical Skills
2. Soft Skills
3. Aptitude Skills
4. Communication Skills

QUAILIFICATIONS

s.no	Class	Grade	Board
1	B.Tech	Running	SOU
2	12 th	B	GSHEB
3	10 th	A	GSHEB

(23)

Aim : Create an HTML Page containing the following Gray Layout using CSS.

Code :

(i)

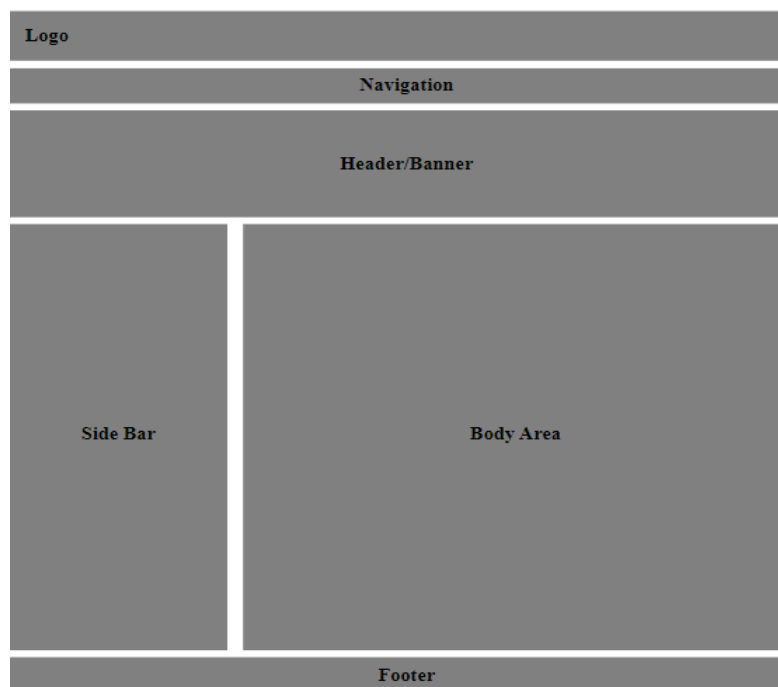
```
<!DOCTYPE html>
  <head>
    <title>Gray Layout</title>
    <style>
      div{font-size: 25px;font-weight: bold;}
      #r1{background-color: gray;height:
50px;width: 98%;padding-top: 20px;
      padding-left: 2%;margin-bottom:
10px;}
      #r2,#r5{background-color: gray;height:
40px;width: 98%;padding-top: 10px;
      padding-left: 2%;margin-bottom:
10px;text-align: center;}
      #r3{background-color: gray;height:
90px;width: 98%;padding-top: 60px;
      padding-left: 2%;margin-bottom:
10px;text-align: center;}
      #r4{height: 600px;width: 100%;margin-
bottom: 10px;}
      #r4 div{float: left;background-color:
gray;padding-top: 280px;height: 320px;
      text-align: center;}
      #r4-c1{width: 28%;margin-right: 2%;}
```

```

        #r4-c2{width: 70%;}
    </style>
</head>
<body>
    <div>
        <div id="r1">Logo</div>
        <div id="r2">Navigation</div>
        <div id="r3">Header/Banner </div>
        <div id="r4">
            <div id="r4-c1">Side Bar</div>
            <div id="r4-c2">Body Area</div>
        </div>
        <div id="r5">Footer</div>
    </div>
</body>
</html>

```

Output :



(ii)

Code :

```
<!DOCTYPE html>

<head>
  <title>
    Gray Layout
  </title>

  <style>
    div{
      font-size: 25px;
      font-weight: bold;
    }
    #logo{
      background-color: gray;
      height: 50px;
      width: 98%;
      padding-top: 20px;
      padding-left: 2%;
      margin-bottom: 10px;
    }
    #Header{

      background-color: gray;
      height: 90px;
      width: 98%;
      padding-top: 60px;
      padding-left: 2%;
      margin-bottom: 10px;
```

```
        text-align: center;
    }
    #Intro{
        background-color: gray;
        height: 70px;
        width: 98%;
        padding-top: 60px;
        padding-left: 2%;
        margin-bottom: 10px;
        text-align: center;
    }
    #box{
        height: 330px;
        width: 100%;
        margin-bottom: 10px;
    }
    #box div{

        float: left;
        background-color: gray;
        padding-top: 150px;
        height: 180px;
        text-align: center;
    }
    #box1{
        margin-right: 2%;
        width: 32%;
    }
    #box2{
        margin-right: 2%;
        width: 32%;
```

```
    }
    #box3{
        width: 32%;

    }
    #footer{
        background-color: gray;
        height: 50px;
        text-align: center;
        padding-top: 10px;
    }
</style>
```

```
</head>
```

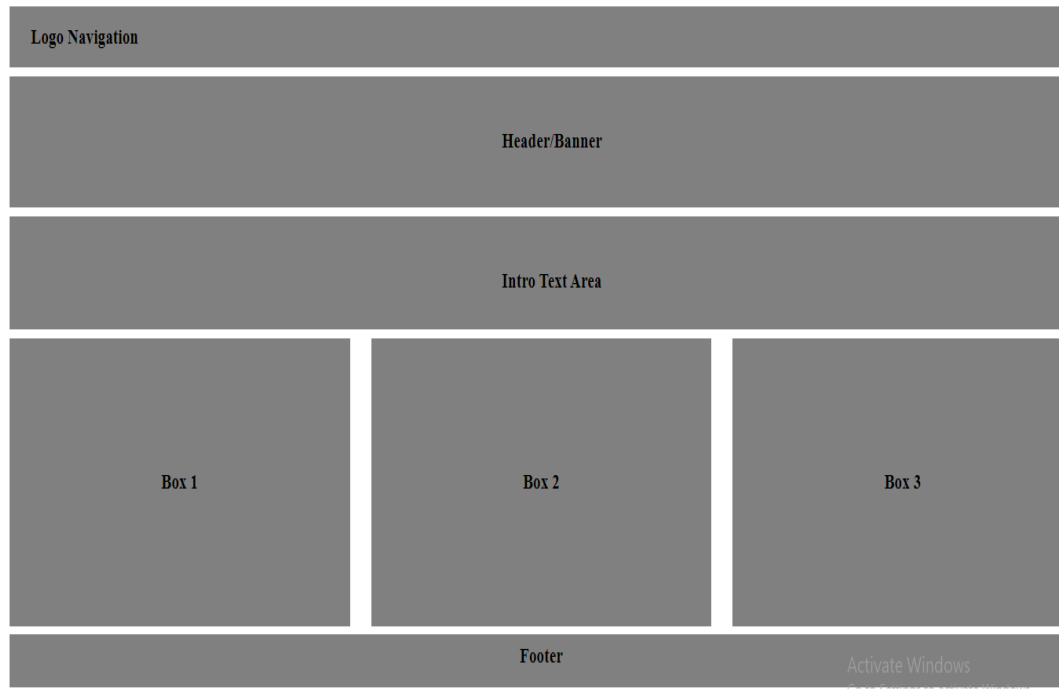
```
<body>
```

```
    <div>
        <div id="logo">Logo Navigation</div>
        <div id="Header">Header/Banner</div>
        <div id="Intro">Intro Text Area</div>
        <div id="box">
            <div id="box1">Box 1</div>
            <div id="box2">Box 2</div>
            <div id="box3">Box 3</div>
        </div>
        <div id="footer">Footer</div>
    </div>
```

```
</body>
```

```
</html>
```

Output :



(iii)

Code :

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">

    <style>
        #r1 {
            padding-top: 30px;
            height: 30px;
            margin-top: 10px;
            margin-left: 10%;
            width: 78%;
```



```
padding-left: 2%;
background-color: gray;
font-size: larger;
}

#r2 div {
float: left;
margin-left: 10%;
margin-top: 10px;
height: 40px;
padding-top: 40px;
text-align:
    center;
}

#r2c1 {
width: 20%;
margin-right: 5%;
font-size: large;
}

#r2c2 {
width: 45%;
background-color: gray;
}

#r3 {
margin-bottom: 10px;
margin-left: 45%;
width: 45%;
height: 250px;
padding-top: 250px;
text-align: center;
background-color: gray;
margin-top: 100px;
```

```

    }

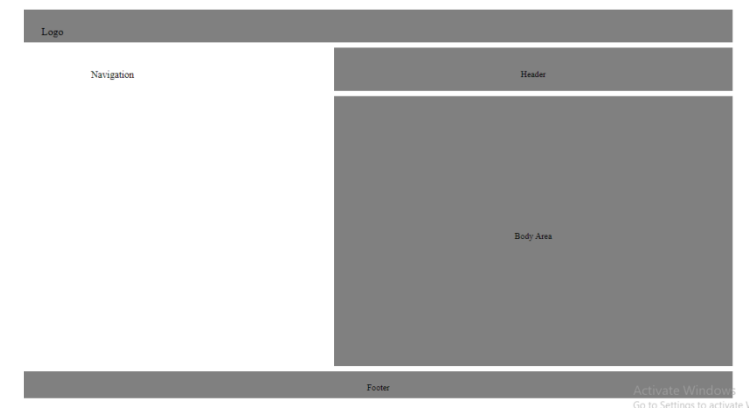
    #r4 {
        margin-left: 10%;
        width: 80%;
        text-align: center;
        padding-top: 20px;
        height: 30px;
        background-color: gray;
    }
</style>
</head>

<body>
    <div id="r1">Logo</div>
    <div id="r2">
        <div id="r2c1">Navigation</div>
        <div id="r2c2">Header</div>
    </div>
    <div id="r3">Body Area</div>
    <div id="r4">Footer</div>
</body>

</html>

```

Output:



(IV.)

Code:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <title>Gray Layout 4</title>
  <style>
    #r1 {
      background-color: gray;
      width: 78%;
      padding-top: 15px;
      padding-left: 2%;
      margin-left: 10%;
      height: 40px;
    }

    #r1c1 {
      float: left;
      font-size: larger;
    }

    #r1c2 {
      margin-left: 80%;
      padding-top: 5px;
    }

    #r3 {
      margin-bottom: 10px;
      height: 100px;
      margin-left: 10%;
      margin-top: 10px;
    }

    #c1,
    #c2,
    #c3 {
```

```
        float: left;
        background-color: gray;
        width: 10%;
        margin-right: 15%;
        text-align:
            center;
    }

    #c5,
    #c6,
    #c7 {
        float: left;
        background-color: gray;
        width: 10%;
        margin-right: 15%;
        text-align: center;
    }

    #c9,
    #c10,
    #c11 {
        float: left;
        background-color: gray;
        width: 10%;
        margin-right: 15%;
        text-align:
            center;
    }

    #c13,
    #c14,
    #c15 {
        float: left;
        background-color: gray;
        width: 10%;
        margin-right: 15%;
        text-align:
            center;
    }
```

```
#r4 div {  
    float: left;  
    background-color: gray;  
    font-weight: bolder;  
}
```

```
#r4c1 {  
    margin-left: 10%;  
    text-align: center;  
    width: 25%;  
    height: 100px;  
    margin-top: 10px;  
    paddingtop: 100px;  
}
```

```
#r4c2 {  
    margin-left: 2.5%;  
    text-align: center;  
    width: 25%;  
    height: 100px;  
    margin-top: 10px;  
    paddingtop: 100px;  
}
```

```
#r4c3 {  
    margin-left: 2.5%;  
    text-align: center;  
    width: 25%;  
    height: 100px;  
    margin-top: 10px;  
    paddingtop: 100px;  
}
```

```
#r5 {  
    width: 80%;  
    margin-left: 10%;  
    background-color: gray;  
    margin-top: 250px;
```

```

        height:
            30px;
        padding-top: 20px;
        text-align: center;
    }
</style>
</head>

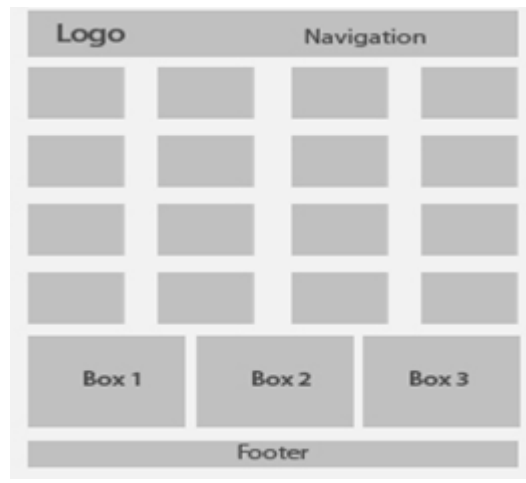
<body>
    <div id="r1">
        <div id="r1c1">Logo</div>
        <div id="r1c2">Navigation</div>
    </div>
    <div>
        <div></div>
    </div>
    <div id="r3">
        <div id="c1">1</div>
        <div id="c2">2</div>
        <div id="c3">3</div>
        <div id="c4">4</div><br>
        <div id="c5">5</div>
        <div id="c6">6</div>
        <div id="c7">7</div>
        <div id="c8">8</div><br>
        <div id="c9">9</div>
        <div id="c10">10</div>
        <div id="c11">11</div>
        <div id="c12">12</div><br>
        <div id="c13">13</div>
        <div id="c14">14</div>
        <div id="c15">15</div>
        <div id="c16">16</div>
    </div>
    <div id="r4">
        <div id="r4c1">Box 1</div>
        <div id="r4c2">Box 2</div>
        <div id="r4c3">Box 3</div>
    </div>

```

```
<div id="r5">Footer</div>
</body>

</html>
```

Output:



(24)

Aim: Demonstrate JavaScript Form Validation with proper examples.

Code:

```
<!DOCTYPE html>
<html>

<head>
  <title>Basic Form Validation</title>
  <script>
    function validateForm() {
      var name = document.forms["myForm"]["name"].value;
      var email = document.forms["myForm"]["email"].value;
      var password = document.forms["myForm"]["password"].value;
```

```

        if (name == "" || email == "" || password == "") {
            alert("Please fill out all fields");
            return false;
        }
    }
</script>
</head>

<body>
    <form name="myForm" onsubmit="return validateForm()">
        <label for="name">Name:</label>
        <input type="text" id="name" name="name">
        <br><br>
        <label for="email">Email:</label>
        <input type="email" id="email" name="email">
        <br><br>
        <label for="password">Password:</label>
        <input type="password" id="password" name="password">
        <br><br>
        <input type="submit" value="Submit">
    </form>
</body>

</html>

```

Output:

Name:

Email:

Password:

127.0.0.1:5500 says
Please fill out all fields

OK

(25)

Aim: Write a javascript to check if the number is even or odd.

Code:


```

<!DOCTYPE html>
<html>

<head>
  <title>Even or Odd Checker</title>
  <script>
    function checkEvenOrOdd() {
      var number = document.getElementById("number").value;
      if (number % 2 == 0) {
        document.getElementById("result").innerHTML = number + " is even";
      } else {
        document.getElementById("result").innerHTML = number + " is odd";
      }
    }
  </script>
</head>

<body>
  <label for="number">Enter a number:</label>
  <input type="number" id="number">
  <br><br>
  <button onclick="checkEvenOrOdd()">Check</button>
  <br><br>
  <div id="result"></div>
</body>

</html>

```

Output:

Enter a number:

3 is odd

(26)

Aim: Create a page and access the LocationAPI

Code:

```
<!DOCTYPE html>
<html>

<head>
  <title>Location API Example</title>
  <script>
    function getLocation() {
      if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(showPosition);
      }
      else {
        alert("Geolocation is not supported by this
browser.");
      }
    }
    function showPosition(position) {
      var latitude = position.coords.latitude;
      var longitude = position.coords.longitude;
      var accuracy = position.coords.accuracy;
      var timestamp = new Date(position.timestamp);
      document.getElementById("latitude").innerHTML = "Latitude:
" + latitude;
      document.getElementById("longitude").innerHTML =
"Longitude: " + longitude;
      document.getElementById("accuracy").innerHTML = "Accuracy:
" + accuracy + " meters";
      document.getElementById("timestamp").innerHTML =
"Timestamp: " + timestamp;
    }
  </script>
</head>

<body>
  <h1>Location API Example</h1>
  <button onclick="getLocation()">Get Location</button>
```

```
<br><br>
<div id="latitude"></div>
<div id="longitude"></div>
<div id="accuracy"></div>
<div id="timestamp"></div>
</body>

</html>
```

Output:

Location API Example

Get Location

Latitude: 21.535707

Longitude: 70.450813

Accuracy: 22 meters

Timestamp: Wed Mar 22 2023 10:14:58 GMT+0530 (India Standard Time)

(27)

Aim: Create a simple XMLHttpRequest, and retrieve the data from the text file.

Code:

```
<!DOCTYPE html>
<html>

<head>
  <title>XMLHttpRequest Example</title>
  <script>
    function loadData() {
      var xhr = new XMLHttpRequest();
```

```

        xhr.open('GET', 'example.txt');
        xhr.onreadystatechange = function () {
            if (xhr.readyState === XMLHttpRequest.DONE) {
                if (xhr.status === 200) {
                    var response = xhr.responseText;
                    document.getElementById('output').innerHTML =
response;
                }
                else {
                    document.getElementById('output').innerHTML =
'Error: ' + xhr.status;
                }
            }
        };
        xhr.send();
    }
</script>
</head>

<body>
    <h1>XMLHttpRequest Example</h1>
    <button onclick="loadData()">Load Data</button>
    <br><br>
    <div id="output"></div>
</body>

</html>

```

Output:

Xml Http Request Example

Error: 404
