|  |  |
| --- | --- |
| **Name:** | **Anas Parekh** |
| **Enrollment No. :** | **2202031000078** |
| **Subject:** | **OJT Practicals** |
| **Branch:** | **B.Tech (IT)** |

**C & C++**

**Practical-1**

**Aim: Write a C program to print the address of a variable using a pointer.**

**Code:**

**#include <stdio.h>**

**int main() {**

**int x = 42;**

**int\* ptr = &x;**

**printf("The address of x is: %p\n", ptr);**

**return 0;**

**}**

**Output:**

**The address of the variable 'num' is 0x7ffd413ca33c**

**Practical-2**

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

**Code:**

**#include <stdio.h>**

**int main() {**

**int num1, num2, result;**

**char op;**

**int\* ptr1 = &num1;**

**int\* ptr2 = &num2;**

**int\* ptrResult = &result;**

**printf("Enter first number: ");**

**scanf("%d", ptr1);**

**printf("Enter operator (+, -, \*, /): ");**

**scanf(" %c", &op);**

**printf("Enter second number: ");**

**scanf("%d", ptr2);**

**switch(op) {**

**case '+':**

**\*ptrResult = \*ptr1 + \*ptr2;**

**break;**

**case '-':**

**\*ptrResult = \*ptr1 - \*ptr2;**

**break;**

**case '\*':**

**\*ptrResult = \*ptr1 \* \*ptr2;**

**break;**

**case '/':**

**\*ptrResult = \*ptr1 / \*ptr2;**

**break;**

**default:**

**printf("Invalid operator");**

**return 1;**

**}**

**printf("Result: %d", \*ptrResult);**

**return 0;**

**}**

**Output:**

**Enter first number: 2**

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

**Enter second number: 3**

**Result: 5**

**Practical-3**

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

**#include <stdio.h>**

**{**

**int temp = x;**

**x = y;**

**y = temp;**

**printf("Values inside swap\_call\_by\_value function:\n");**

**printf("x = %d, y = %d\n", x, y);**

**}**

**void swap\_call\_by\_reference(int\* x, int\* y)**

**{**

**int temp = \*x;**

**\*x = \*y;**

**\*y = temp;**

**printf("Values inside swap\_call\_by\_reference function:\n");**

**printf("x = %d, y = %d\n", \*x, \*y);**

**}**

**int main()**

**{**

**int a = 10, b = 20;**

**printf("Before swapping:\n");**

**printf("a = %d, b = %d\n", a, b);**

**swap\_call\_by\_value(a, b);**

**printf("After swapping using call by value:\n");**

**printf("a = %d, b = %d\n\n", a, b);**

**printf("Before swapping:\n");**

**printf("a = %d, b = %d\n", a, b);**

**swap\_call\_by\_reference(&a, &b);**

**printf("After swapping using call by reference:\n");**

**printf("a = %d, b = %d\n", a, b);**

**return 0;**

**}**

**Output:**

**Before swapping:**

**a = 10, b = 20**

**Values inside swap\_call\_by\_value function:**

**x = 20, y = 10**

**After swapping using call by value:**

**a = 10, b = 20**

**Before swapping:**

**a = 10, b = 20**

**Values inside swap\_call\_by\_reference function:**

**x = 20, y = 10**

**After swapping using call by reference:**

**a = 20, b = 10**

**Practical-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 <stdlib.h>**

**#include <string.h>**

**#include <time.h>**

**struct personal {**

**char name[50];**

**int birth\_year;**

**int birth\_month;**

**int birth\_day;**

**int age;**

**};**

**void calculate\_age(struct personal \*person) {**

**time\_t t = time(NULL);**

**struct tm tm = \*localtime(&t);**

**int current\_year = tm.tm\_year + 1900;**

**int current\_month = tm.tm\_mon + 1;**

**int current\_day = tm.tm\_mday;**

**person->age = current\_year - person->birth\_year;**

**if (current\_month < person->birth\_month ||**

**(current\_month == person->birth\_month && current\_day < person->birth\_day)) {**

**person->age--;**

**}**

**}**

**int main() {**

**struct personal people[4];**

**for (int i = 0; i < 4; i++) {**

**printf("Enter person %d's name: ", i+1);**

**fgets(people[i].name, 50, stdin);**

**people[i].name[strcspn(people[i].name, "\n")] = 0; // removing trailing newline character**

**printf("Enter person %d's birth year (yyyy): ", i+1);**

**scanf("%d", &people[i].birth\_year);**

**printf("Enter person %d's birth month (mm): ", i+1);**

**scanf("%d", &people[i].birth\_month);**

**printf("Enter person %d's birth day (dd): ", i+1);**

**scanf("%d", &people[i].birth\_day);**

**getchar(); // consume newline character**

**calculate\_age(&people[i]);**

**}**

**for (int i = 0; i < 4; i++) {**

**printf("\nPerson %d:\n", i+1);**

**printf("Name: %s\n", people[i].name);**

**printf("Date of Birth: %02d/%02d/%d\n", people[i].birth\_day, people[i].birth\_month, people[i].birth\_year);**

**printf("Age: %d\n", people[i].age);**

**}**

**return 0;**

**}**

**Output:**

**Enter person 1's name: abc**

**Enter person 1's birth year (yyyy): 2000**

**Enter person 1's birth month (mm): 1**

**Enter person 1's birth day (dd): 1**

**Enter person 2's name: pqr**

**Enter person 2's birth year (yyyy): 2001**

**Enter person 2's birth month (mm): 2**

**Enter person 2's birth day (dd): 2**

**Enter person 3's name: xyz**

**Enter person 3's birth year (yyyy): 2002**

**Enter person 3's birth month (mm): 3**

**Enter person 3's birth day (dd): 3**

**Enter person 4's name: efg**

**Enter person 4's birth year (yyyy): 2003**

**Enter person 4's birth month (mm): 3**

**Enter person 4's birth day (dd): 3**

**Person 1:**

**Name: abc**

**Date of Birth: 01/01/2000**

**Age: 23**

**Person 2:**

**Name: pqr**

**Date of Birth: 02/02/2001**

**Age: 22**

**Person 3:**

**Name: xyz**

**Date of Birth: 03/03/2002**

**Age: 21**

**Person 4:**

**Name: efg**

**Date of Birth: 03/03/2003**

**Age: 20**

**Practical-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, sum = 0;**

**printf("Enter the number of elements: ");**

**scanf("%d", &n);**

**int\* nums = (int\*) malloc(n \* sizeof(int)); // allocate memory dynamically**

**if (nums == NULL) { // check if allocation failed**

**printf("Memory allocation failed!");**

**return 1;**

**}**

**printf("Enter %d integers:\n", n);**

**for (int i = 0; i < n; i++) {**

**scanf("%d", &nums[i]);**

**sum += nums[i];**

**}**

**printf("The sum is: %d\n", sum);**

**free(nums); // deallocate memory**

**return 0;**

**Output:**

**Enter the number of elements: 5**

**Enter 5 integers:**

**1 2 3 4 5**

**The sum is: 15**

**Practical-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, \*odd\_fp, \*even\_fp;**

**int num;**

**fp = fopen("New", "r");**

**if (fp == NULL) {**

**printf("Error opening file New!");**

**return 1;**

**}**

**odd\_fp = fopen("odd", "w");**

**if (odd\_fp == NULL) {**

**printf("Error opening file odd!");**

**return 1;**

**}**

**even\_fp = fopen("even", "w");**

**if (even\_fp == NULL) {**

**printf("Error opening file even!");**

**return 1;**

**}**

**while (fscanf(fp, "%d", &num) != EOF) {**

**if (num % 2 == 0) {**

**// write even numbers to even file**

**fprintf(even\_fp, "%d\n", num);**

**} else {**

**// write odd numbers to odd file**

**fprintf(odd\_fp, "%d\n", num);**

**}**

**}**

**fclose(fp);**

**fclose(odd\_fp);**

**fclose(even\_fp);**

**printf("Odd numbers:\n");**

**odd\_fp = fopen("odd", "r");**

**if (odd\_fp == NULL) {**

**printf("Error opening file odd!");**

**return 1;**

**}**

**while (fscanf(odd\_fp, "%d", &num) != EOF) {**

**printf("%d\n", num);**

**}**

**fclose(odd\_fp);**

**// print the values of even file**

**printf("\nEven numbers:\n");**

**even\_fp = fopen("even", "r");**

**if (even\_fp == NULL) {**

**printf("Error opening file even!");**

**return 1;**

**}**

**while (fscanf(even\_fp, "%d", &num) != EOF) {**

**printf("%d\n", num);**

**}**

**fclose(even\_fp);**

**return 0;**

**}**

**Output:**

**Practical-7**

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

**Code:**

**#include <iostream>**

**using namespace std;**

**bool isPrime(int num);**

**int main() {**

**int num;**

**cout << "Enter a positive integer: ";**

**cin >> num;**

**if (isPrime(num)) {**

**cout << num << " is a prime number." << endl;**

**} else {**

**cout << num << " is not a prime number." << endl;**

**}**

**return 0;**

**}**

**bool isPrime(int num) {**

**if (num <= 1) {**

**return false;**

**}**

**for (int i = 2; i <= num / 2; i++) {**

**if (num % i == 0) {**

**return false;**

**}**

**}**

**return true;**

**}**

**Output:**

**Enter a positive integer: 4**

**4 is not a prime number.**

**Practical-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;**

**int main() {**

**char letter;**

**cout << "Enter a letter: ";**

**cin >> letter;**

**if (letter == 'a' || letter == 'e' || letter == 'i' || letter == 'o' || letter == 'u' ||**

**letter == 'A' || letter == 'E' || letter == 'I' || letter == 'O' || letter == 'U') {**

**cout << letter << " is a vowel." << endl;**

**} else if ((letter >= 'a' && letter <= 'z') || (letter >= 'A' && letter <= 'Z')) {**

**cout << letter << " is a consonant." << endl;**

**} else {**

**cout << letter << " is not a letter." << endl;**

**}**

**return 0;**

**}**

**Output:**

**Enter a letter: c**

**c is a consonant.**

**Practical-9**

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

**Code:**

**#include <iostream>**

**using namespace std;**

**class MyClass {**

**public:**

**MyClass() {**

**cout << "Constructor called." << endl;**

**}**

**~MyClass() {**

**cout << "Destructor called." << endl;**

**}**

**};**

**int main() {**

**cout << "Creating an object..." << endl;**

**MyClass obj;**

**cout << "Exiting the program..." << endl;**

**return 0;**

**}**

**Output:**

**Creating an object...**

**Constructor called.**

**Exiting the program...**

**Destructor called.**

**Practical-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 {**

**protected:**

**int roll\_no;**

**string name;**

**public:**

**void input() {**

**cout << "Enter roll number: ";**

**cin >> roll\_no;**

**cout << "Enter name: ";**

**cin >> name;**

**}**

**void display() {**

**cout << "Roll number: " << roll\_no << endl;**

**cout << "Name: " << name << endl;**

**}**

**};**

**class Test {**

**protected:**

**int marks[5];**

**public:**

**void input() {**

**cout << "Enter marks in five subjects: ";**

**for (int i = 0; i < 5; i++) {**

**cin >> marks[i];**

**}**

**}**

**void display() {**

**cout << "Marks in five subjects: ";**

**for (int i = 0; i < 5; i++) {**

**cout << marks[i] << " ";**

**}**

**cout << endl;**

**}**

**};**

**class Result : public Student, public Test {**

**protected:**

**int total\_marks;**

**float percentage;**

**public:**

**void calculate() {**

**total\_marks = 0;**

**for (int i = 0; i < 5; i++) {**

**total\_marks += marks[i];**

**}**

**percentage = (float)total\_marks / 5.0;**

**}**

**void display() {**

**Student::display();**

**Test::display();**

**cout << "Total marks: " << total\_marks << endl;**

**cout << "Percentage: " << percentage << "%" << endl;**

**}**

**};**

**int main() {**

**Result res;**

**res.input();**

**res.calculate();**

**res.display();**

**return 0;**

**}**

**Practical-11**

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

**Code:**

**#include <iostream>**

**using namespace std;**

**class MyClass {**

**public:**

**int value;**

**MyClass() {}**

**MyClass(int v) { value = v; }**

**MyClass operator+ (const MyClass& obj) {**

**MyClass res;**

**res.value = value + obj.value;**

**return res;**

**}**

**};**

**int main() {**

**MyClass a(10);**

**MyClass b(20);**

**MyClass c = a + b;**

**cout << "a + b = " << c.value << endl;**

**return 0;**

**}**

**Output:**

**a + b = 30**

**Practical-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 width, height;**

**public:**

**void get\_data() {**

**cout << "Enter width and height: ";**

**cin >> width >> height;**

**}**

**virtual void display\_area() = 0; // Pure virtual function**

**};**

**class TRIANGLE : public SHAPE {**

**public:**

**void display\_area() {**

**double area = 0.5 \* width \* height;**

**cout << "Area of triangle: " << area << endl;**

**}**

**};**

**class RECTANGLE : public SHAPE {**

**public:**

**void display\_area() {**

**double area = width \* height;**

**cout << "Area of rectangle: " << area << endl;**

**}**

**};**

**int main() {**

**SHAPE \*s;**

**TRIANGLE t;**

**s = &t;**

**s->get\_data();**

**s->display\_area();**

**RECTANGLE r;**

**s = &r;**

**s->get\_data();**

**s->display\_area();**

**return 0;**

**}**

**Output:**

**Enter width and height: 12 13**

**Area of triangle: 78**

**Enter width and height: 34 11**

**Area of rectangle: 374**

**DBMS:**

**Practical-13**

**Aim: To study DDL-create and DML-insert commands.**

**Create following Table**

**Code:**

**CREATE TABLE Job (**

**job\_id VARCHAR(15),**

**job\_title VARCHAR(30),**

**min\_sal INT,**

**max\_sal INT**

**);**

**CREATE TABLE Employee (**

**emp\_no INT,**

**emp\_name VARCHAR(30),**

**emp\_sal DECIMAL(8,2),**

**emp\_comm DECIMAL(6,1),**

**dept\_no INT**

**);**

**CREATE TABLE Deposit (**

**a\_no INT IDENTITY(1,1),**

**cname VARCHAR(50),**

**bname VARCHAR(30),**

**amount DECIMAL(4,2),**

**a\_date DATE**

**);**

**CREATE TABLE Borrow (**

**loanno INT,**

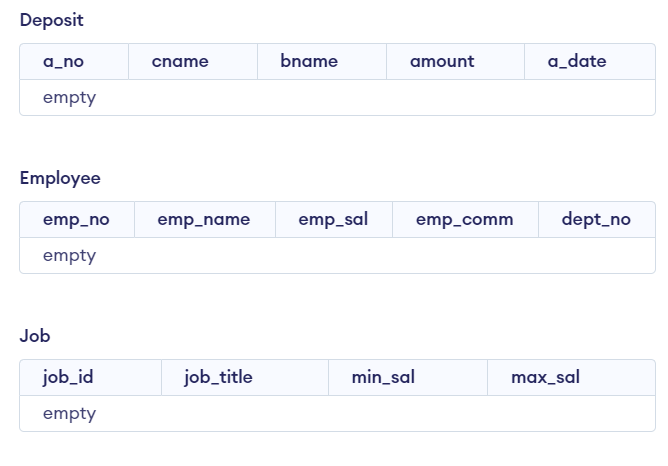
**cname VARCHAR(25),**

**bname VARCHAR(20),**

**amount DECIMAL(6,2)**

**);**

**Output:**

****

**Practical-15**

**Aim: Create tables and insert sample data in tables.**

**Write SQL queries to insert following data into tables**

**Code:**

**INSERT INTO Employee (emp\_n, emp\_name, emp\_sal, emp\_comm, dept\_no)**

**VALUES**

**(101, 'Smith', 800, 20, 20),**

**(102, 'Snehal', 1600, 300, 25),**

**(103, 'Adama', 1100, 0, 20),**

**(104, 'Aman', 3000, 15, NULL),**

**(105, 'Anita', 5000, 50000, 10),**

**(106, 'Sneha', 2450, 24500, 10),**

**(107, 'Anamika', 2975, 30, NULL);**

**INSERT INTO Job (job\_id, job\_name, min\_sal, max\_sal)**

**VALUES**

**('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);**

**INSERT INTO deposit (A\_no, cname, Bname, Amount, date)**

**VALUES**

**(101, 'Anil', 'andheri', 7000, '2006-01-01'),**

**(102, 'sunil', 'virar', 5000, '2006-07-15'),**

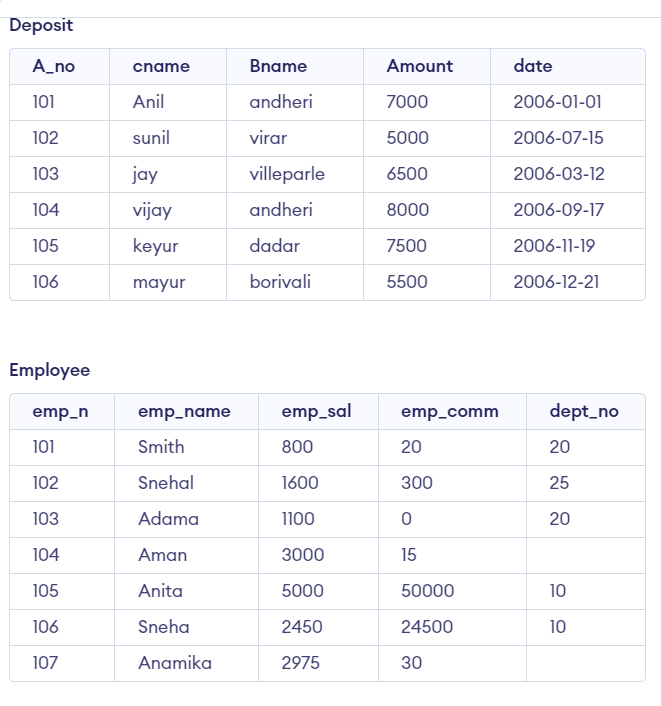
**(103, 'jay', 'villeparle', 6500, '2006-03-12'),**

**(104, 'vijay', 'andheri', 8000, '2006-09-17'),**

**(105, 'keyur', 'dadar', 7500, '2006-11-19'),**

**(106, 'mayur', 'borivali', 5500, '2006-12-21');**

**Output:**

****

**Practical-17**

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

**1. List total deposit from deposit. => SELECT SUM(Amount) as Total\_Deposit FROM deposit;**

**2. List total amount from andheri branch =>**

**SELECT SUM(Amount) as Total\_Amount FROM deposit WHERE Bname='andheri';**

**3. Count total number of customers =>**

**SELECT COUNT(\*) as Total\_Customers FROM deposit;**

**4. Count total number of customer’s cities. =>**

**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’. =>**

**UPDATE Employee SET dept\_no=10 WHERE SUBSTRING(emp\_name, 2, 1)='m';**

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

**UPDATE Employee SET emp\_name='Adam' WHERE emp\_n=103;**

**7. Write a query to display the current date. Label the column Date => SELECT CURDATE() as 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 =>**

**SELECT emp\_n, emp\_sal, ROUND(emp\_sal\*1.15) 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. =>**

**SELECT emp\_n, emp\_sal, ROUND(emp\_sal\*1.15) as New\_Salary, ROUND(emp\_sal\*0.15) as Increment FROM Employee;**

**Practical-18**

**Aim:** **Write the SQL queries to perform numeric, date and String functions.**

1. **Retrieve all data from employee, jobs and deposit. =>**

**SELECT \* FROM Employee;**

**SELECT \* FROM Job;**

**SELECT \* FROM deposit**

1. **Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06. => SELECT A\_no, Amount, date FROM deposit**

**WHERE date BETWEEN '2006-01-01' AND '2006-07-25'**

1. **Display all jobs with minimum salary is greater than 4000. => SELECT \* FROM Job WHERE min\_sal > 4000;**
2. **Display name and salary of employee whose department no is 20. Give alias name to name of employee. =>**

**SELECT emp\_name AS name, emp\_sal AS salary FROM Employee WHERE dept\_no = 20;**

1. **Display employee no,name and department details of those employee whose department lies in(10,20) =>**

**SELECT emp\_n, emp\_name, dept**

1. **Display all employee whose name start with ‘A’ and third character is ‘ ‘a’. =>**

**SELECT \* FROM Employee WHERE emp\_name LIKE 'A\_a%';**

1. **Display name, number and salary of those employees whose name is 5 characters long and first three characters are ‘Ani’. =>**

**SELECT emp\_name, emp\_n, emp\_sal FROM Employee WHERE emp\_name LIKE 'Ani\_\_';**

1. **Display the non-null values of employees and also employee name second charactershould be ‘n’ and string should be 5 character long. =>**

**SELECT \* FROM Employee WHERE emp\_name LIKE 'n\_\_' AND emp\_sal IS NOT NULL AND emp\_comm IS NOT NULL AND dept\_no IS NOT NULL;**

1. **Display the null values of employee and also employee name’s third character should be ‘a’. =>**

**SELECT \* FROM Employee WHERE emp\_name LIKE '\_\_a%' AND (emp\_sal IS NULL OR emp\_comm IS NULL OR dept\_no IS NULL);**

**# HTML, CSS & JavaScript:**

**Practical-18**

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

**Code:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Resume</title>**

**</head>**

**<body>**

**<header>**

**<h1>ABC</h1>**

**<p>Web Developer</p><hr>**

**</header>**

**<main>**

**<section>**

**<h2>Contact Information</h2>**

**<ul>**

**<li>Email: abc@gmail.com</li>**

**<li>Phone: 1234567890</li>**

**<li>Address: Earth</li>**

**</ul>**

**</section><hr>**

**<section>**

**<h2>Education</h2>**

**<h3>Bachelor In Technology</h3>**

**<p>XYZ University</p>**

**<p>Graduated: May 20XX</p>**

**</section><hr>**

**<section>**

**<h2>Skills</h2>**

**<ul>**

**<li>HTML</li>**

**<li>CSS</li>**

**<li>JavaScript</li>**

**<li>PHP</li>**

**<li>MySQL</li>**

**</ul>**

**</section><hr>**

**<section>**

**<h2>Experience</h2>**

**<h3>Web Developer</h3>**

**<p>ABC Company</p>**

**<p>June 20XX - Present</p>**

**<h3>Web Design Intern</h3>**

**<p>XYZ Company</p>**

**<p>Summer 20XX</p>**

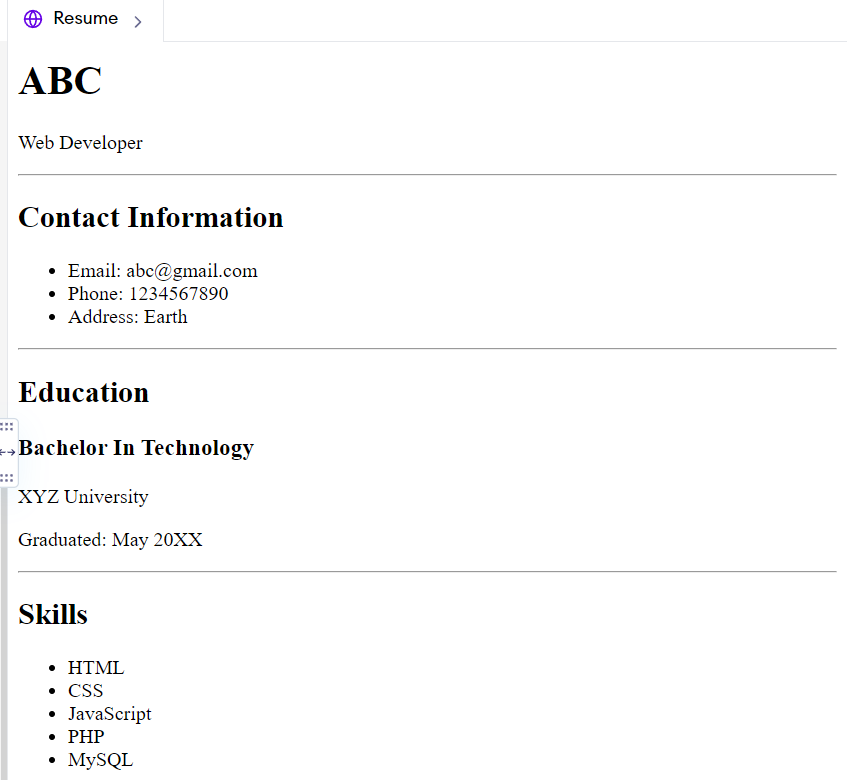
**</section><hr>**

**</main>**

**</body>**

**</html>**

**Output:**

****

**Practical-19**

**Aim: 19. Create an HTML webpage that shows Poster Presentation using all Table Properties**

**Code:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Poster Presentation</title>**

**</head>**

**<body>**

**<h1>Poster Presentation</h1>**

**<table border="1" cellpadding="10" cellspacing="0">**

**<tr>**

**<th colspan="2">Introduction</th>**

**</tr>**

**<tr>**

**<td rowspan="2">ABC</td>**

**<td>ABC</td>**

**</tr>**

**<tr>**

**<td>ABC</td>**

**</tr>**

**<tr>**

**<th colspan="2">Methodology</th>**

**</tr>**

**<tr>**

**<td>ABC</td>**

**<td>ABC</td>**

**</tr>**

**<tr>**

**<td>ABC </td>**

**<td>ABC</td>**

**</tr>**

**<tr>**

**<td>ABC </td>**

**<td>ABC</td>**

**</tr>**

**<tr>**

**<th colspan="2">Results</th>**

**</tr>**

**<tr>**

**<td>ABC</td>**

**<td>ABC</td>**

**</tr>**

**<tr>**

**<td>ABC</td>**

**<td>ABC</td>**

**</tr>**

**<tr>**

**<th colspan="2">Conclusion</th>**

**</tr>**

**<tr>**

**<td colspan="2">Summary of Poster presentation</td>**

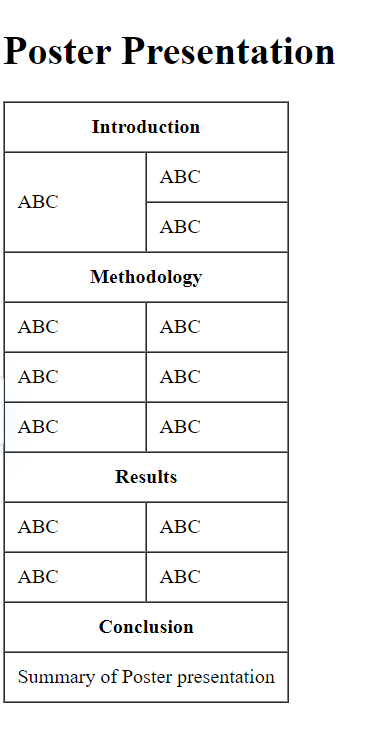
**</tr>**

**</table>**

**</body>**

**</html>**

**Output:**

****

**Practical-20**

**Aim: Create an HTML page table and form.**

**Code:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<title>Table and Form Example</title>**

**</head>**

**<body>**

**<h1>Table Example</h1>**

**<table border="1">**

**<thead>**

**<tr>**

**<th>Column 1</th>**

**<th>Column 2</th>**

**<th>Column 3</th>**

**</tr>**

**</thead>**

**<tbody>**

**<tr>**

**<td>Row 1, Column 1</td>**

**<td>Row 1, Column 2</td>**

**<td>Row 1, Column 3</td>**

**</tr>**

**<tr>**

**<td>Row 2, Column 1</td>**

**<td>Row 2, Column 2</td>**

**<td>Row 2, Column 3</td>**

**</tr>**

**<tr>**

**<td>Row 3, Column 1</td>**

**<td>Row 3, Column 2</td>**

**<td>Row 3, Column 3</td>**

**</tr>**

**</tbody>**

**</table>**

**<h1>Form Example</h1>**

**<form action="#" method="POST">**

**<label for="name">Name:</label>**

**<input type="text" id="name" name="name"><br>**

**<label for="email">Email:</label>**

**<input type="email" id="email" name="email"><br>**

**<label for="message">Message:</label>**

**<textarea id="message" name="message"></textarea><br>**

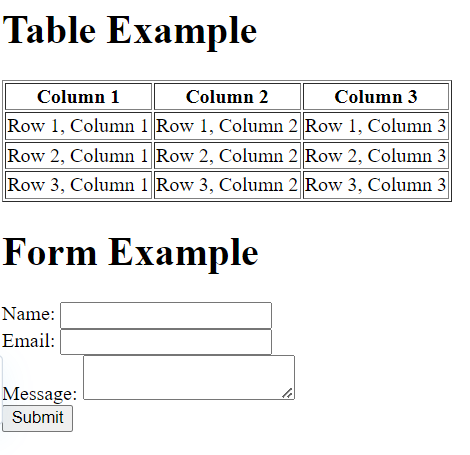
**<input type="submit" value="Submit">**

**</form>**

**</body>**

**</html>**

**Output:**

****

**Practical-21**

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

**Code:**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<title>Registration Form</title>**

**</head>**

**<body>**

**<h1>Registration Form</h1>**

**<form method="post">**

**<label for="name">Name:</label>**

**<input type="text" id="name" name="name" required minlength="2" maxlength="50"><br><br>**

**<label for="email">Email:</label>**

**<input type="email" id="email" name="email" required><br><br>**

**<label for="phone">Phone:</label>**

**<input type="tel" id="phone" name="phone" pattern="[0-9]{10}" required><br><br>**

**<label for="password">Password:</label>**

**<input type="password" id="password" name="password" required minlength="8"><br><br>**

**<label for="confirm\_password">Confirm Password:</label>**

**<input type="password" id="confirm\_password" name="confirm\_password" required minlength="8" onchange="validatePassword()"><br><br>**

**<input type="submit" value="Submit">**

**</form>**

**<script>**

**function validatePassword() {**

**const password = document.getElementById("password");**

**const confirm\_password = document.getElementById("confirm\_password");**

**if (password.value != confirm\_password.value) {**

**confirm\_password.setCustomValidity("Passwords do not match");**

**} else {**

**confirm\_password.setCustomValidity("");**

**}**

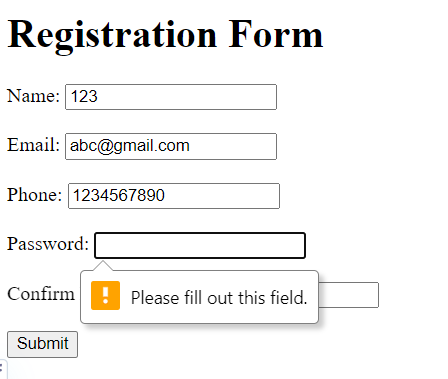
**}**

**</script>**

**</body>**

**</html>**

**Output:**

****

**Practical-22**

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

**Code:**

**<html>**

**<head>**

**<title>My Resume</title>**

**<style>**

**body {**

**font-family: Arial, sans-serif;**

**background-color: #f2f2f2;**

**margin: 0;**

**padding: 0;**

**}**

**.container {**

**max-width: 800px;**

**margin: 0 auto;**

**padding: 20px;**

**background-color: #fff;**

**border: 1px solid #ccc;**

**}**

**h1, h2 {**

**margin: 0;**

**text-align: center;**

**}**

**h1 {**

**font-size: 36px;**

**color: #333;**

**margin-bottom: 20px;**

**}**

**h2 {**

**font-size: 24px;**

**color: #666;**

**margin-top: 20px;**

**}**

**ul {**

**list-style-type: none;**

**margin: 0;**

**padding: 0;**

**}**

**li {**

**margin-bottom: 10px;**

**}**

**.contact {**

**margin-top: 30px;**

**display: flex;**

**justify-content: space-between;**

**align-items: center;**

**background-color: #f2f2f2;**

**padding: 10px 20px;**

**}**

**.contact a {**

**color: #333;**

**text-decoration: none;**

**}**

**.contact a:hover {**

**color: #666;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<h1>ABC</h1>**

**<h2>Front-end Developer</h2>**

**<ul>**

**<li><strong>Email:</strong> abc@email.com</li>**

**<li><strong>Phone:</strong> 1234567890</li>**

**<li><strong>Address:</strong>Earth**

**</ul>**

**<h2>Skills</h2>**

**<ul>**

**<li>HTML</li>**

**<li>CSS</li>**

**<li>JavaScript</li>**

**<li>React</li>**

**<li>Bootstrap</li>**

**</ul>**

**<h2>Experience</h2>**

**<ul>**

**<li><strong>Front-end Developer</strong> | ABC Company | June 2018 - Present</li>**

**<li><strong>Web Designer</strong> | XYZ Company | January 2017 - June 2018</li>**

**</ul>**

**<div class="contact">**

**<p>Connect with me:</p>**

**<a href="#">LinkedIn</a>**

**<a href="#">Twitter</a>**

**<a href="#">GitHub</a>**

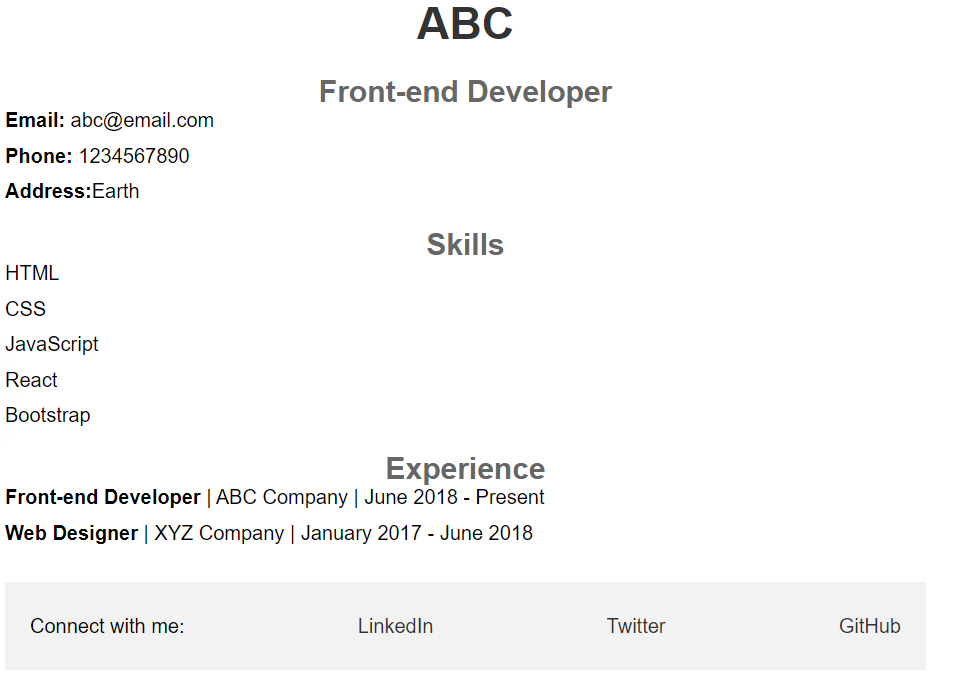
**</div>**

**</div>**

**</body>**

**</html>**

**Output:**

****

**Practical-23**

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

**Code:**

**Layout-1:**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Grid Layout 1</title>**

**<style>**

**.container{**

**display: grid;**

**gap: 10px;**

**background-color: lightgray;**

**padding: 10px;**

**}**

**.grid-item{**

**background-color: cornflowerblue;**

**text-align: center;**

**font-size: 20px;**

**}**

**.item1{**

**padding: 30px;**

**grid-column: 1 / span 5;**

**grid-row: 1;**

**}**

**.item2{**

**padding: 10px;**

**grid-column: 1 / span 5;**

**grid-row: 2;**

**}**

**.item3{**

**grid-column: 1 / span 5;**

**grid-row: 3;**

**padding: 40px;**

**}**

**.item4{**

**padding: 100px;**

**grid-column: 1 / span 2;**

**grid-row: 4;**

**grid-column-start: 1;**

**grid-column-end: 2;**

**}**

**.item5{**

**grid-column: 3 / span 3;**

**grid-row: 4;**

**grid-column-start: 2;**

**grid-column-end: 6;**

**padding: 170px;**

**}**

**.item6{**

**grid-column: 1 / span 5;**

**grid-row: 6;**

**padding: 10px;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<div class="grid-item item1">Logo</div>**

**<div class="grid-item item2">Navigation</div>**

**<div class="grid-item item3">Header</div>**

**<div class="grid-item item4">Side Bar</div>**

**<div class="grid-item item5">Body Area</div>**

**<div class="grid-item item6">Footer</div>**

**</div>**

**</body>**

**</html>**

**Layout-2:**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Grid Layout 2</title>**

**<style>**

**.container{**

**display: grid;**

**gap: 10px;**

**background-color: lightgray;**

**padding: 10px;**

**padding-left: 300px;**

**padding-right: 300px;**

**}**

**.grid-item{**

**background-color: cornflowerblue;**

**text-align: center;**

**font-size: 20px;**

**}**

**.item1{**

**padding: 30px;**

**grid-column: 1 / span 5;**

**grid-row: 1;**

**}**

**.item2{**

**padding: 40px;**

**grid-column: 1 / span 2;**

**grid-row: 2 / span 3;**

**background-color:rgb(227, 227, 212);**

**}**

**.item3{**

**grid-column: 3 / span 3;**

**grid-row: 2;**

**padding: 40px;**

**}**

**.item5{**

**grid-column: 3 / span 3;**

**grid-row: 4;**

**/\* grid-column-start: 3;**

**grid-column-end: 6; \*/**

**padding: 170px;**

**}**

**.item6{**

**grid-column: 1 / span 5;**

**grid-row: 6;**

**padding: 10px;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<div class="grid-item item1">Logo</div>**

**<div class="grid-item item2">Side bar Navigation</div>**

**<div class="grid-item item3">Header</div>**

**<div class="grid-item item5">Body Area</div>**

**<div class="grid-item item6">Footer</div>**

**</div>**

**</body>**

**</html>**

**Layout-3:**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Grid Layout 3</title>**

**<style>**

**.container{**

**display: grid;**

**gap: 10px;**

**background-color: lightgray;**

**padding: 10px;**

**padding-left: 200px;**

**padding-right: 200px;**

**}**

**.grid-item{**

**background-color: rgb(119, 113, 239);**

**text-align: center;**

**font-size: 20px;**

**}**

**.box-item1{**

**padding: 20px;**

**grid-column: 1 / span 6;**

**grid-row: 1;**

**}**

**.box-item2{**

**padding: 40px;**

**grid-column: 1 / span 6;**

**grid-row: 2;**

**}**

**.box-item3{**

**padding: 40px;**

**grid-column: 1 / span 6;**

**grid-row: 3;**

**}**

**.box-item4{**

**padding: 220px;**

**grid-column: 1 / span 2;**

**grid-row: 4;**

**}**

**.box-item5{**

**padding: 220px;**

**grid-column: 3 / span 2;**

**grid-row: 4;**

**}**

**.box-item6{**

**padding: 220px;**

**grid-column: 5 / span 2;**

**grid-row: 4;**

**}**

**.box-item7{**

**padding: 20px;**

**grid-column: 1 / span 6;**

**grid-row: 5;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<div class="grid-item box-item1">Logo</div>**

**<div class="grid-item box-item2">Header</div>**

**<div class="grid-item box-item3">Text Area</div>**

**<div class="grid-item box-item4">Box 1</div>**

**<div class="grid-item box-item5">Box 2</div>**

**<div class="grid-item box-item6">Box 3</div>**

**<div class="grid-item box-item7">Footer</div>**

**</div>**

**</body>**

**</html>**

**Layout-4:**

**<!DOCTYPE html>**

**<html lang="en">**

**<head>**

**<meta charset="UTF-8">**

**<meta http-equiv="X-UA-Compatible" content="IE=edge">**

**<meta name="viewport" content="width=device-width, initial-scale=1.0">**

**<title>Grid Layout 4</title>**

**<style>**

**.container{**

**display: grid;**

**gap: 10px;**

**background-color: lightgray;**

**padding: 10px;**

**padding-left: 200px;**

**padding-right: 200px;**

**}**

**.grid-item{**

**background-color: rgb(119, 113, 239);**

**text-align: center;**

**font-size: 20px;**

**}**

**.box-item{**

**background-color: rgb(238, 93, 77);**

**padding: 20px;**

**column-gap: 10px;**

**gap: 10px;**

**}**

**.logo{**

**grid-column: 1 / span 4;**

**grid-row: 1;**

**padding: 25px;**

**}**

**.navi{**

**grid-column: 4 / span 5;**

**grid-row: 1;**

**padding: 25px;**

**}**

**#item1{**

**grid-column: 1 / span 2;**

**grid-row: 2;**

**}**

**#item2{**

**grid-column: 3 / span 2;**

**grid-row: 2;**

**}**

**#item3{**

**grid-column: 5 / span 2;**

**grid-row: 2;**

**}**

**#item4{**

**grid-column: 7 / span 2;**

**grid-row: 2;**

**}**

**#item5{**

**grid-column: 1 / span 2;**

**grid-row: 3;**

**}**

**#item6{**

**grid-column: 3 / span 2;**

**grid-row: 3;**

**}**

**#item7{**

**grid-column: 5 / span 2;**

**grid-row: 3;**

**}**

**#item8{**

**grid-column: 7 / span 2;**

**grid-row: 3;**

**}**

**#item9{**

**grid-column: 1 / span 2;**

**grid-row: 4;**

**}**

**#item10{**

**grid-column: 3 / span 2;**

**grid-row: 4;**

**}**

**#item11{**

**grid-column: 5 / span 2;**

**grid-row: 4;**

**}**

**#item12{**

**grid-column: 7 / span 2;**

**grid-row: 4;**

**}**

**#item13{**

**grid-column: 1 / span 2;**

**grid-row: 5;**

**}**

**#item14{**

**grid-column: 3 / span 2;**

**grid-row: 5;**

**}**

**#item15{**

**grid-column: 5 / span 2;**

**grid-row: 5;**

**}**

**#item16{**

**grid-column: 7 / span 2;**

**grid-row: 5;**

**}**

**.box1{**

**grid-column: 1 / span 2;**

**grid-row: 6;**

**padding: 90px;**

**background-color: rgba(42, 173, 42, 0.742);**

**}**

**.box2{**

**grid-column: 4 / span 2;**

**grid-row: 6;**

**padding: 90px;**

**background-color: rgba(42, 173, 42, 0.742);**

**}**

**.box3{**

**grid-column: 7 / span 2;**

**grid-row: 6;**

**padding: 90px;**

**background-color: rgba(42, 173, 42, 0.742);**

**}**

**.footer{**

**grid-column: 1 / span 8;**

**grid-row: 7;**

**padding: 15px;**

**background-color: rgb(119, 113, 239);**

**}**

**.box-item{**

**height: 3px;**

**}**

**</style>**

**</head>**

**<body>**

**<div class="container">**

**<div class="grid-item logo">Logo</div>**

**<div class="grid-item navi">Navigation</div>**

**<div class="box-item" id="item1"></div>**

**<div class="box-item" id="item2"></div>**

**<div class="box-item" id="item3"></div>**

**<div class="box-item" id="item4"></div>**

**<div class="box-item" id="item5"></div>**

**<div class="box-item" id="item6"></div>**

**<div class="box-item" id="item7"></div>**

**<div class="box-item" id="item8"></div>**

**<div class="box-item" id="item9"></div>**

**<div class="box-item" id="item10"></div>**

**<div class="box-item" id="item11"></div>**

**<div class="box-item" id="item12"></div>**

**<div class="box-item" id="item13"></div>**

**<div class="box-item" id="item14"></div>**

**<div class="box-item" id="item15"></div>**

**<div class="box-item" id="item16"></div>**

**<div class="grid-item box1">Box 1</div>**

**<div class="grid-item box2">Box 2</div>**

**<div class="grid-item box3">Box 3</div>**

**<div class="grid-item footer">Footer</div>**

**</div>**

**</body>**

**</html>**

**Practical-24**

**Aim: Demonstrate JavaScript Form Validation with proper examples.**

**Code:**

**<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="text" id="email" name="email"><br><br>**

**<input type="submit" value="Submit">**

**</form>**

**function validateForm() {**

**var name = document.forms["myForm"]["name"].value;**

**var email = document.forms["myForm"]["email"].value;**

**if (name == "") {**

**alert("Name must be filled out");**

**return false;**

**}**

**if (email == "") {**

**alert("Email must be filled out");**

**return false;**

**}**

**if (!validateEmail(email)) {**

**alert("Invalid email address");**

**return false;**

**}**

**}**

**function validateEmail(email) {**

**var re = /\S+@\S+\.\S+/;**

**return re.test(email);**

**}**

**Practical-25**

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

**Code:**

**function checkEvenOrOdd(num) {**

**if (num % 2 === 0) {**

**return num + " is even";**

**} else {**

**return num + " is odd";**

**}**

**}**

**Output:**

**console.log(checkEvenOrOdd(4)); // Output: 4 is even**

**console.log(checkEvenOrOdd(7)); // Output: 7 is odd**

**Practical-26**

**Aim: Create a page and access the LocationAPI.**

**Code:**

**<!DOCTYPE html>**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>Location API Example</title>**

**</head>**

**<body>**

**<h1>Location API Example</h1>**

**<p>Open the console to see the geolocation data.</p>**

**<script>**

**if (navigator.geolocation) {**

**navigator.geolocation.getCurrentPosition(function(position) {**

**console.log("Latitude: " + position.coords.latitude);**

**console.log("Longitude: " + position.coords.longitude);**

**});**

**} else {**

**console.log("Geolocation is not supported by this browser.");**

**}**

**</script>**

**</body>**

**</html>**

**Output:**

**Latitude: 37.7749**

**Longitude: -122.4194**

**Practical-27**

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

**Code:**

**<html>**

**<head>**

**<meta charset="UTF-8">**

**<title>XMLHTTPRequest Example</title>**

**</head>**

**<body>**

**<h1>XMLHTTPRequest Example</h1>**

**<p id="demo"></p>**

**<script>**

**var xhttp = new XMLHttpRequest();**

**xhttp.onreadystatechange = function() {**

**if (this.readyState == 4 && this.status == 200) {**

**document.getElementById("demo").innerHTML = this.responseText;**

**}**

**};**

**xhttp.open("GET", "example.txt", true);**

**xhttp.send();**

**</script>**

**</body>**

**</html>**

**Output: This is some example text in a text file.**