|  |  |  |
| --- | --- | --- |
|  | INDEX |  |
|  |  |  |
| S. No. | Title Of Experiment | Page No. |
|  | **PART A- Programs in Java** |  |
| 1 | How to Use This Manual…………. | 2 |
|  | Vision Mission of institute | 5 |
|  | Vision Mission of Department | 5 |
|  | Program Outcomes | 6 |
| 2 | Lab Prerequisites | 8 |
| 2.1 | Laboratory Objective | 8 |
| 2.2 | Laboratory Equipment |  |
| 2.3 | Write a Java Program to Check Even or Odd Number | 10 |
| 2.4 | Write a program to find Fibonacci series of a given no. | 12 |
| 2.5 | Write a Java program that works as a simple calculator. | 14 |
| 2.6 | Develop an applet that displays a simple message. | 16 |
| 2.7 | Write a Java program that handles all mouse events | 20 |
| 2.8 | Implementation of Inheritance. | 22 |
| 2.9 | Implementation of multithreading. | 24 |
| 2.10 | Write a Java program that reads a line of integers and then displays each integer and the sum of all integers.(classes). | 26 |
| 2.11 | Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations. | 28 |
|  | **Part B- Web Technology (Website Design using (HTML5))** |  |
| 2.12 | (Table Formatting): Design a mark sheet and display all your marks with subjects in a tabular format using table. | 30 |
| 2.13 | Write a program to demonstrate Event Handling -Validation of JSPM’s RAJARSHI SHAHU COLLEGE OF ENGINEERING TATHAWADE, PUNE-33 Autonomous Institute Affiliated to Savitribai Phule Pune University Structure and Syllabi of S. Y. B. Tech IT Page 49 of 49 registration form -Open a Window from the current window -Change color of background at each click of button or refresh of a page -Display calendar for the month and year selected from combo box On Mouse over event | 34 |
| 2.14 | (JavaScript) Validate the Registration, user login, user profile and payment by credit card pages using JavaScript. | 36 |
| 2.15 | (JavaScript) Using javascript demonstrate Nested loop. | 38 |
| 2.16 | (database ) Write a suitable script which, inserts, 5update, deletes and display records from the database. | 40 |

**Vision Mission PSOs of Department**

**Vision**

“To create quality information technology professionals through superior academic environment.”

**Mission**

* To incorporate the IT fundamentals in students to be successful in their career.
* To motivate students for higher studies, research and entrepreneurship.
* To provide IT services to society.

**Program Outcomes (POs)**

Engineering Graduates will be able to:

**1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

**2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

**3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

**4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

**5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

**6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice. JSPM’s Rajarshi Shahu College of Engineering Department of IT Engineering

**7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

**8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

**9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

**10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

**11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

**12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**Lab Prerequisites :** Basic knowledge of Computer Programming, Knowledge of any object- oriented language like C++ is helpful but not mandatory

**Lab Objectives:** To familiarize the students with fundamentals of object oriented programming. The aim is to make the students develop desktop and web applications using Java programming.

**Laboratory Equipment**

**Hardware and Software required:**

* A working computer system with either Windows or Linux
* A web browser either IE or firefox
* Tomcat web server and Apache web server
* A database either Mysql or Oracle
* JVM(Java virtual machine) must be installed on your system

**PART A- Programs in Java**

**Assignment 1**

**Title** : Write a Java Program to Check Even or Odd Number

**Objective**:

In this program, we learn to check if a number entered by an user is even or odd. This will be done using if...else statement and ternary operator in Java.

**Theory:**

In this program, we have an int variable num. We are using [**Scanner class**](https://beginnersbook.com/2022/08/java-scanner-class-with-examples/) to get the number entered by the user.

Once the entered number is stored in the variable num, we are using [**if..else statement**](https://beginnersbook.com/2017/08/if-else-statement-in-java/) to check if the number is **perfectly divisible by 2 or not**. Based on the outcome, it is executing if block (if condition true) or else block (if condition is false).

**Program:**

import java.util.Scanner;

public class JavaExample

{

public static void main(String args[])

{

int num;

System.out.print("Enter an Integer number: ");

//The input provided by user is stored in num

Scanner input = new Scanner(System.in);

num = input.nextInt();

// If number is divisible by 2 then it's an even number

//else it is an odd number

if ( num % 2 == 0 )

System.out.println(num+" is an even number.");

else

System.out.println(num+" is an odd number.");

}

}

**Output:**

Enter an Integer number: 27

27 is an odd number

**Conclusion:**

Thus we have implemented a program to find wheatear given number is even or odd.

**Assignment 2**

**Title:** Write a program to find Fibonacci series of a given no.

**Objective**: This JAVA program is to find fibonacci series upto a given range. Fibonacci series is a series in which each number is the sum of preceding two numbers.For example, fibonacci series upto n=7 will be 0,1,1,2,3,5.

**Program:**

import java.util.Scanner;

public class JavaExample

{

public static void main(String[] args)

{

int num, num1 = 0, num2 = 1;

System.out.print("Enter an Integer number: ");

//The input provided by user is stored in num

Scanner input = new Scanner(System.in);

num = input.nextInt();

System.out.print("Fibonacci Series of "+num+" numbers:");

for (int i = 1; i <= num; ++i)

{

System.out.print(num1+" ");

/\* On each iteration, we are assigning second number

\* to the first number and assigning the sum of last two

\* numbers to the second number

\*/

int sumOfPrevTwo = num1 + num2;

num1 = num2;

num2 = sumOfPrevTwo;

}

}

}

**Output:**

Fibonacci Series of 7 numbers:0 1 1 2 3 5 8

## Conclusion :

Thus we have implemented a program to find Fibonacci series of a given no.

**Assignment 3**

**Title** : Write a Java program that works as a simple calculator.

**Objective**: In this program, you'll learn to make a simple calculator using switch..case in Java. This calculator would be able to add, subtract, multiply and divide two numbers.

**Program:**

import java.util.Scanner;

class Main {

public static void main(String[] args) {

char operator;

Double number1, number2, result;

// create an object of Scanner class

Scanner input = new Scanner(System.in);

// ask users to enter operator

System.out.println("Choose an operator: +, -, \*, or /");

operator = input.next().charAt(0);

// ask users to enter numbers

System.out.println("Enter first number");

number1 = input.nextDouble();

System.out.println("Enter second number");

number2 = input.nextDouble();

switch (operator) {

// performs addition between numbers

case '+':

result = number1 + number2;

System.out.println(number1 + " + " + number2 + " = " + result);

break;

// performs subtraction between numbers

case '-':

result = number1 - number2;

System.out.println(number1 + " - " + number2 + " = " + result);

break;

// performs multiplication between numbers

case '\*':

result = number1 \* number2;

System.out.println(number1 + " \* " + number2 + " = " + result);

break;

// performs division between numbers

case '/':

result = number1 / number2;

System.out.println(number1 + " / " + number2 + " = " + result);

break;

default:

System.out.println("Invalid operator!");

break;

}

input.close();

}

}

**Output1 :**

Choose an operator: +, -, \*, or /

\*

Enter first number

3

Enter second number

9

3.0 \* 9.0 = 27.

**Output 2**

Choose an operator: +, -, \*, or /

+

Enter first number

21

Enter second number

8

21.0 + 8.0 = 29.0

**Output3**

Choose an operator: +, -, \*, or /

-

Enter first number

9

Enter second number

3

9.0 - 3.0 = 6.0

**Output 4**

Choose an operator: +, -, \*, or /

/

Enter first number

24

Enter second number

8

24.0 / 8.0 = 3.0

**Conclusion:**

Thus we have implemented a program that works as a simple calculator.

**Assignment 4**

**Title:** Develop an applet that displays a simple message.

**Objective:** An applet is a Java program that can be embedded into a web page. It runs inside the web browser and works at client side. An applet is embedded in an HTML page using the APPLET or OBJECT tag and hosted on a web server.  
Applets are used to make the website more dynamic and entertaining. In this program, you'll learn how to make an applet Program that displays a simple message.

**:**

\* Develop a applet to display the simple message \*/

import java.applet.Applet;  
import java.awt.Graphics;  
public class Hello extends Applet  
{  
public void paint(Graphics g)  
{  
g.drawString("Hello world",50,30);  
}  
}

**Output:**



**Conclusion:**

Thus we have implemented a program that display “Hello World” using applet program.

**Assignment 5**

**Title:** Implementation of Inheritance using JAVA.

**Objective:** Inheritance is one of the key features of OOP that allows us to create a new class from an existing class. The new class that is created is known as **subclass** (child or derived class) and the existing class from where the child class is derived is known as **super class** (parent or base class).The extends keyword is used to perform inheritance in Java. In this program, you'll learn how to implement inheritance program using JAVA.

Program:

class Calculation

{

int z;

public void addition(int x, int y)

{

z = x + y;

System.out.println("The sum of the given numbers:"+z);

}

public void Subtraction(int x, int y)

{

z = x - y;

System.out.println("The difference between the given numbers:"+z);

}

}

public class My\_Calculation extends Calculation {

public void multiplication(int x, int y)

{

z = x \* y;

System.out.println("The product of the given numbers:"+z);

}

public static void main(String args[])

{

int a = 20, b = 10;

My\_Calculation demo = new My\_Calculation();

demo.addition(a, b);

demo.Subtraction(a, b);

demo.multiplication(a, b);

}

}

**Output :**

The sum of the given numbers:30

The difference between the given numbers:10

The product of the given numbers:200

**Conclusion :**

Thus we have implemented a program of single level inheritance using JAVA.

**Assignment 6**

**Title** : Write a Java program that handles all mouse events .

**Objective:** The Java MouseListener is notified whenever you change the state of mouse. It is notified against MouseEvent. The MouseListener interface is found in java.awt.event package. It has five methods.

Methods of MouseListener interface

The signature of 5 methods found in MouseListener interface are given below:

**public** **abstract** **void** mouseClicked(MouseEvent e);

**public** **abstract** **void** mouseEntered(MouseEvent e);

**public** **abstract** **void** mouseExited(MouseEvent e);

**public** **abstract** **void** mousePressed(MouseEvent e);

**public** **abstract** **void** mouseReleased(MouseEvent e);

In this program, you'll learn how to handle mouse events using JAVA.

**Program:**

import javax.swing.\*;

import java.awt.\*;

import javax.swing.event.\*;

import java.awt.event.\*;

class MouseEventPerformer extends JFrame implements MouseListener

{

JLabel l1;

public MouseEventPerformer()

{

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setSize(300,300);

setLayout(new FlowLayout(FlowLayout.CENTER));

l1 = new JLabel();

Font f = new Font("Verdana", Font.BOLD, 20);

l1.setFont(f);

l1.setForeground(Color.BLUE);

add(l1);

addMouseListener(this);

setVisible(true);

}

public void mouseExited(MouseEvent m)

{

l1.setText("Mouse Exited");

}

public void mouseEntered(MouseEvent m)

{

l1.setText("Mouse Entered");

}

public void mouseReleased(MouseEvent m)

{

l1.setText("Mouse Released");

}

public void mousePressed(MouseEvent m)

{

l1.setText("Mouse Pressed");

}

public void mouseClicked(MouseEvent m)

{

l1.setText("Mouse Clicked");

}

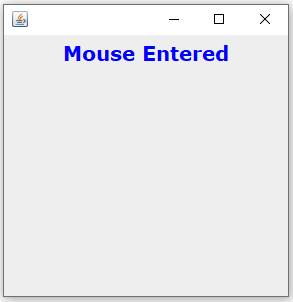
public static void main(String[] args) {

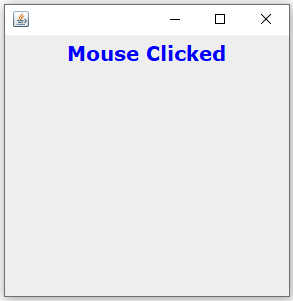
MouseEventPerformer mep = new MouseEventPerformer();

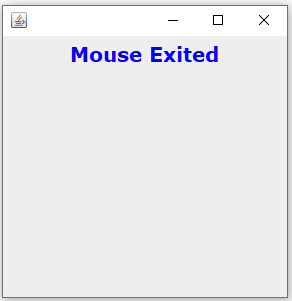
}

}

**Output:**







**Conclusion**: Thus we have implemented a JAVA program to handle mouse events.

**Assignment 7**

**Title** : Implementation of multithreading.

**Objective:**

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.

Threads can be created by using two mechanisms :

1. Extending the Thread class
2. Implementing the Runnable Interface

**Thread creation by extending the Thread class**

We create a class that extends the **java.lang.Thread** class. This class overrides the run() method available in the Thread class. A thread begins its life inside run() method. We create an object of our new class and call start() method to start the execution of a thread. Start() invokes the run() method on the Thread object.

In this program, you'll learn how to implement multithreading using JAVA.

**Program:**

class MultithreadingDemo extends Thread {

    public void run()

    {

        try {

            // Displaying the thread that is running

            System.out.println(

                "Thread " + Thread.currentThread().getId()

                + " is running");

        }

        catch (Exception e) {

            // Throwing an exception

            System.out.println("Exception is caught");

        }

    }

}

// Main Class

public class Multithread {

    public static void main(String[] args)

    {

        int n = 8; // Number of threads

        for (int i = 0; i < n; i++) {

            MultithreadingDemo object

                = new MultithreadingDemo();

            object.start();

        }

    }

}

**Output:**

Thread 15 is running

Thread 14 is running

Thread 16 is running

Thread 12 is running

Thread 11 is running

Thread 13 is running

Thread 18 is running

Thread 17 is running

**Conclusion:** Thus we have implemented a JAVA program for multithreading.

**Assignment 8**

**Title** : Write a Java program that reads a line of integers and then displays each integer and the sum of all integers.(classes).

**Objective:** Java program that reads a line of integers, and displays each integer, and the sum of all the integers ( uses StringTokenizer class of java.util). In this program, you'll learn how to implement multithreading using JAVA.

**Program:**

import java.util.\*;

class StringTokenizerDemo {

public static void main(String args[]) {

int n;

int sum = 0;

Scanner sc = new Scanner(System.in);

System.out.println("Enter integers with one space gap:");

String s = sc.nextLine();

StringTokenizer st = new StringTokenizer(s, " ");

while (st.hasMoreTokens()) {

String temp = st.nextToken();

n = Integer.parseInt(temp);

System.out.println(n);

sum = sum + n;

}

System.out.println("sum of the integers is: " + sum);

sc.close();

}

}

**OUTPUT:**

Enter integers with one space gap:

10 20 30 40 50

10

20

30

40

50

sum of the integers is: 150

**Conclusion:** Thus we have implement a Java program that reads a line of integers and then displays each integer and the sum of all integers.

**Assignment 9**

**Title:** Write a java program that connects to a database using JDBC and does add, delete, modify and retrieve operations.

**Objective:**   
In this program we perform the insert, update and delete operation on a table by using java program. The target database system is Oracle.

**Implementing Insert Statement**

|  |
| --- |
| // Java program to illustrate  // inserting to the Database  import java.sql.\*;  public class insert1  {  public static void main(String args[])      {          String id = "id1";   String pwd = "pwd1";          String fullname = "geeks for geeks";          String email = "geeks@geeks.org"; try          {              Class.forName("oracle.jdbc.driver.OracleDriver");              Connection con=DriverManager.getConnection(" jdbc:oracle:thin:@localhost:1521:orcl","login1","pwd1");              Statement stmt = con.createStatement();                 // Inserting data in database       String q1 = "insert into userid values('" +id+ "', '" +pwd+   "', '" +fullname+ "', '"+email+"')";              int x = stmt.executeUpdate(q1);              if (x > 0)                  System.out.println("Successfully Inserted");  else  System.out.println("Insert Failed");                 con.close();          }          catch(Exception e)          {    System.out.println(e);          }      }  } |
|  |

**Output :**

Successfully Registered

|  |
| --- |
| **Implementing Update Statement**  import java.sql.\*;  public class update1  {      public static void main(String args[])      {          String id = "id1";          String pwd = "pwd1";          String newPwd = "newpwd";          try          {              Class.forName("oracle.jdbc.driver.OracleDriver");              Connection con = DriverManager.getConnection("           jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");              Statement stmt = con.createStatement();                // Updating database         String q1 = "UPDATE userid set pwd = '" + newPwd +                     "' WHERE id = '" +id+ "' AND pwd = '" + pwd + "'";              int x = stmt.executeUpdate(q1);                if (x > 0)                  System.out.println("Password Successfully Updated");              else                  System.out.println("ERROR OCCURRED :(");                con.close();          }          catch(Exception e)          {              System.out.println(e);          }      }  } |

**Output :**

Password Successfully Updated

**Implementing Delete Statement**

|  |
| --- |
| import java.sql.\*;    public class delete  {      public static void main(String args[])      {          String id = "id2";          String pwd = "pwd2";          try          {              Class.forName("oracle.jdbc.driver.OracleDriver");              Connection con = DriverManager.getConnection("               jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");              Statement stmt = con.createStatement();                // Deleting from database              String q1 = "DELETE from userid WHERE id = '" + id +                      "' AND pwd = '" + pwd + "'";                int x = stmt.executeUpdate(q1);                if (x > 0)                  System.out.println("One User Successfully Deleted");              else                  System.out.println("ERROR OCCURRED :(");                con.close();          }          catch(Exception e)          {              System.out.println(e);          }      }  } |

**Output :**

One User Successfully Deleted

**Implementing Select Statement**

|  |
| --- |
| // Java program to illustrate  // selecting from Database  import java.sql.\*;    public class select  {      public static void main(String args[])      {          String id = "id1";          String pwd = "pwd1";          try          {              Class.forName("oracle.jdbc.driver.OracleDriver");              Connection con = DriverManager.getConnection("                      jdbc:oracle:thin:@localhost:1521:orcl", "login1", "pwd1");              Statement stmt = con.createStatement();                // SELECT query              String q1 = "select \* from userid WHERE id = '" + id +                                      "' AND pwd = '" + pwd + "'";              ResultSet rs = stmt.executeQuery(q1);              if (rs.next())              {                  System.out.println("User-Id : " + rs.getString(1));                  System.out.println("Full Name :" + rs.getString(3));                  System.out.println("E-mail :" + rs.getString(4));              }              else              {                  System.out.println("No such user id is already registered");              }              con.close();          }          catch(Exception e)          {              System.out.println(e);          }      }  } |

Output :

User-Id : id1

Full Name : geeks for geeks

E-mail :geeks@geeks.org

**Conclusion:** Thus we have implement a java program that connects to a database using JDBC

|  |
| --- |
| and does add, delete, modify and retrieve operations. |

**Part B- Web Technology (Website Design using (HTML5))**

**Title** : (Table Formatting): Design a mark sheet and display all your marks with subjects in a tabular format using table.

[**index.html**](https://gist.github.com/HackyCoder0951/41f81e418407991437fc60ab2fd14a7b#file-index-html)

|  |  |
| --- | --- |
| <!DOCTYPE html> | |
|  | | <html> | |
|  | | <head> | |
|  | | <title>MARKS SHEET USING HTML TABLES</title> | |
|  | | <meta charset="UTF-8"> | |
|  | | <meta name="viewport" content="width=device-width, initial-scale=1"> | |
|  | | <meta name="Keywords" content="html, css, html tables, table"> | |
|  | | <meta name="Description" content="html table"> | |
|  | | <!-- add icon --> | |
|  | | <link rel="icon" href="/favicon.ico" type="image/x-icon"> | |
|  | | <link href='http://fonts.googleapis.com/css?family=Lato:400,700' rel='stylesheet' type='text/css'> | |
|  | |  | |
|  | | </head> | |
|  | | <body> | |
|  | | <div class="container"> | |
|  | | <h2>HTML TABLE</h2> | |
|  | | <table> | |
|  | | <thead> | |
|  | | <tr> | |
|  | | <th>Roll No.</th> | |
|  | | <th>Name</th> | |
|  | | <th>English</th> | |
|  | | <th>Maths</th> | |
|  | | <th>Science</th> | |
|  | | <th>Computer Science</th> | |
|  | | <th>Social Studies</th> | |
|  | | <th>Total</th> | |
|  | | <th>Max Marks</th> | |
|  | | <th>Grade </th> | |
|  | | <tr> | |
|  | | </thead> | |
|  | | <tbody> | |
|  | | <tr> | |
|  | | <td>01</td> | |
|  | | <td>Bhasker</td> | |
|  | | <td>86</td> | |
|  | | <td>77</td> | |
|  | | <td>87</td> | |
|  | | <td>92</td> | |
|  | | <td>95</td> | |
|  | | <td>437</td> | |
|  | | <td>500</td> | |
|  | | <td>A</td> | |
|  | | </tr> | |
|  | | <tr> | |
|  | | <td>02</td> | |
|  | | <td>Balwant</td> | |
|  | | <td>86</td> | |
|  | | <td>77</td> | |
|  | | <td>87</td> | |
|  | | <td>92</td> | |
|  | | <td>95</td> | |
|  | | <td>437</td> | |
|  | | <td>500</td> | |
|  | | <td>A</td> | |
|  | | </tr> | |
|  | | <tr> | |
|  | | <td>03</td> | |
|  | | <td>Kailash</td> | |
|  | | <td>86</td> | |
|  | | <td>77</td> | |
|  | | <td>87</td> | |
|  | | <td>92</td> | |
|  | | <td>95</td> | |
|  | | <td>437</td> | |
|  | | <td>500</td> | |
|  | | <td>A</td> | |
|  | | </tr> | |
|  | | <tr> | |
|  | | <td>04</td> | |
|  | | <td>Nikhil</td> | |
|  | | <td>86</td> | |
|  | | <td>77</td> | |
|  | | <td>87</td> | |
|  | | <td>92</td> | |
|  | | <td>95</td> | |
|  | | <td>437</td> | |
|  | | <td>500</td> | |
|  | | <td>A</td> | |
|  | | </tr> | |
|  | | <tr> | |
|  | | <td>05</td> | |
|  | | <td>Shubham</td> | |
|  | | <td>86</td> | |
|  | | <td>77</td> | |
|  | | <td>87</td> | |
|  | | <td>92</td> | |
|  | | <td>95</td> | |
|  | | <td>437</td> | |
|  | | <td>500</td> | |
|  | | <td>A</td> | |
|  | | </tr> | |
|  | | </tbody> | |
|  | | </table> | |
|  | | </div> | |
|  | | </body> | |
|  | | </html>  [**style.css**](https://gist.github.com/HackyCoder0951/41f81e418407991437fc60ab2fd14a7b#file-style-css) | |
| body { |
|  | background-color: #c5cae9; | |
|  | padding: 25px; | |
|  | } | |
|  |  | |
|  | .container { | |
|  | width: 800px; | |
|  | height: 440px; | |
|  | margin: 0 auto; | |
|  | padding-left: 32px; | |
|  | padding-right: 32px; | |
|  | padding-top: 40px; | |
|  | border-radius: 12px; | |
|  | background-color: #90a4ae; | |
|  | font-family: Lato; | |
|  | } | |
|  |  | |
|  | .container h2 { | |
|  | text-align: center; | |
|  | } | |
|  |  | |
|  | table { | |
|  | margin: 0 auto; | |
|  | } | |
|  |  | |
|  | td, | |
|  | th { | |
|  | padding: 12px; | |
|  | border: 2px dotted; | |
|  | } | |

<!DOCTYPE html>

<html>

<head>

<meta charset= "utf-8" >

<title>This is an example for nested loop in JavaScript</title>

</head>

<body>

<p>Click below button to loop inner loop each ( 5 ) times for outer loop.</p>

<button onclick="myFunction()">Click Here</button>

<p id="did"></p>

<script>

function myFunction() {

var text = "";

var i;

var j;

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

for (j = 0; j < 2; j++) {

text += "The number is i = " + i + " and j = " + j + "<br>";

}

}

document.getElementById("did").innerHTML = text;

}

</script>

</body>

</html>

**Output:**

**<script>**

function validateform(){

var name=document.myform.name.value;

var password=document.myform.password.value;

if (name==null || name==""){

  alert("Name can't be blank");

  return false;

}else if(password.length**<6**){

  alert("Password must be at least 6 characters long.");

  return false;

  }

}

**</script>**

**<body>**

**<form** name="myform" method="post" action="abc.jsp" onsubmit="return validateform()" **>**

Name: **<input** type="text" name="name"**><br/>**

Password: **<input** type="password" name="password"**><br/>**

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

**</form>**

<html>

    <head>

        <script>

            function GEEKFORGEEKS() {

                var name =

                    document.forms.RegForm.Name.value;

                var email =

                    document.forms.RegForm.EMail.value;

                var phone =

                    document.forms.RegForm.Telephone.value;

                var what =

                    document.forms.RegForm.Subject.value;

                var password =

                    document.forms.RegForm.Password.value;

                var address =

                    document.forms.RegForm.Address.value;

                var regEmail=/^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/g;  //Javascript reGex for Email Validation.

                var regPhone=/^\d{10}$/;                                        // Javascript reGex for Phone Number validation.

                var regName = /\d+$/g;                                    // Javascript reGex for Name validation

                if (name == "" || regName.test(name)) {

                    window.alert("Please enter your name properly.");

                    name.focus();

                    return false;

                }

                if (address == "") {

                    window.alert("Please enter your address.");

                    address.focus();

                    return false;

                }

                if (email == "" || !regEmail.test(email)) {

                    window.alert("Please enter a valid e-mail address.");

                    email.focus();

                    return false;

                }

                if (password == "") {

                    alert("Please enter your password");

                    password.focus();

                    return false;

                }

                if(password.length <6){

                    alert("Password should be atleast 6 character long");

                    password.focus();

                    return false;

                }

                if (phone == "" || !regPhone.test(phone)) {

                    alert("Please enter valid phone number.");

                    phone.focus();

                    return false;

                }

                if (what.selectedIndex == -1) {

                    alert("Please enter your course.");

                    what.focus();

                    return false;

                }

                return true;

            }

        </script>

        <style>

            div {

                box-sizing: border-box;

                width: 100%;

                border: 100px solid black;

                float: left;

                align-content: center;

                align-items: center;

            }

            form {

                margin: 0 auto;

                width: 600px;

            }

        </style>

    </head>

    <body>

        <h1 style="text-align: center;">REGISTRATION FORM</h1>

        <form name="RegForm" onsubmit="return GEEKFORGEEKS()" method="post">

<p>Name: <input type="text"

                            size="65" name="Name" /></p>

            <br />

<p>Address: <input type="text"

                            size="65" name="Address" />

        </p>

            <br />

<p>E-mail Address: <input type="text"

                            size="65" name="EMail" /></p>

            <br />

<p>Password: <input type="text"

                        size="65" name="Password" /></p>

            <br />

<p>Telephone: <input type="text"

                        size="65" name="Telephone" /></p>

            <br />

<p>

                SELECT YOUR COURSE

                <select type="text" value="" name="Subject">

                    <option>BTECH</option>

                    <option>BBA</option>

                    <option>BCA</option>

                    <option>B.COM</option>

                    <option>GEEKFORGEEKS</option>

                </select>

            </p>

            <br />

            <br />

<p>Comments: <textarea cols="55"

                            name="Comment"> </textarea></p>

<p>

                <input type="submit"

                    value="send" name="Submit" />

                <input type="reset"

                    value="Reset" name="Reset" />

            </p>

        </form>

    </body>

</html>