

1. Write a program to find the factorial value of my number entered by the user. using javascript.

Example output:

inp:- 3 o/p:- 6
 3 24
 4 120
 5

Ans

Program:

<!DOCTYPE HTML>

<html>

<head>

<script>

function fact() {

var n;

n = Number(document.getElementById("factorial").value);

var i, ans = 1;

for (i = 1; i <= n; i++) {

ans *= i;

}

document.getElementById("answer").value = ans;

}

</script>

</head>

<body>

Enter any number: <input id = "factorial">

<button onclick = "fact()"> calculate factorial </button>

<input id = "answer">

</body>

</html>

Output:

Enter any number : 5

Calculate factorial

120

Company policy: If an employee's salary is less than 1500/-, then HRA = 10% of BS and DA = 90% of BS. If his BS is either equal to or above 1500/-, then employee's salary is HRA = 500/- & DA = 98% of BS. If the Employee's Salary is input, write a program to find his gross salary using JS Eventhandlers

Ans # Program

```
<!DOCTYPE HTML>
```

```
<html>
```

```
<head>
```

```
<script>
```

```
function calc() {
```

```
var n, ans;
```

```
n = Number(document.getElementById("salary").value);
```

```
if (n < 1500) {
```

```
ans = n + ((10/100) * n) + ((90/100) * n);
```

```
} else {
```

```
ans = n + 500 + (0.98 * n);
```

```
}
```

```
document.getElementById("answer").value = ans;
```

```
}
```

```
</script>
```

```
</head>
```

```
<body>
```

```
Enter Salary: <input id = "salary">
```

```
<button onclick = "calc()"> Gross Salary </button> <br />
```

```
<input id = "answer">
```

```
</body>
```

```
</html>
```

Output:

Enter Salary :

Gross Salary

3. Write a program to take a character (c) as input and check whether the given character is a vowel or a consonant. Using Java.

Ans

Program.

```
import java.util.Scanner;
```

```
Public class VowelOrNot {
```

```
    Public static void main(String args[]) {
```

```
        char c;
```

```
        Scanner sc = new Scanner(System.in);
```

```
        c = sc.next().charAt(0);
```

```
        if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') {
```

```
            System.out.println("Vowel");
```

```
        } else {
```

```
            System.out.println("Consonant");
```

```
        }
```

```
    }
```

```
}
```

Output 1:-

IP :- A

Vowel

Output 2:-

IP :- C

Consonant

Ans: 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 thread 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 thread object.

~~class~~ # Program

```
class Multithreading extends Thread {  
    public void run() {  
        System.out.println ("Thread " + Thread.currentThread().getId() +  
                             " is running");  
    }  
}
```

```
public class Multithread {  
    public static void main(String[] args) {  
        int n = 3;  
        for (int i = 0; i < n; i++) {  
            Multithreading object = new Multithreading();  
            object.start();  
        }  
    }  
}
```

Output:

```
Thread 3 is running  
Thread 4 is running  
Thread 5 is running
```

Thread creation by implementing the Runnable Interface.

We create a new class which implements java.lang.Runnable interface and override run() method. Then we instantiate a Thread object and call start() method on this object.

Program.

```
class Multithreading implements Runnable {  
    public void run() {  
        System.out.println("Thread " + Thread.currentThread().getId() +  
            " is running");  
    }  
}  
  
class Multi {  
    public static void main (String [] args) {  
        int n = 3;  
        for (int i = 0; i < n; i++) {  
            Thread obj = new Thread(new Multithreading());  
            obj.start();  
        }  
    }  
}
```

Output:

```
Thread 3 is running  
Thread 4 is running  
Thread 5 is running.
```


5. Connect to the database using any of the JDBC drivers. Insert and display 2 records with Bus Ticket Reservation information like Ticket (8 digit no.), Source, destination, date of Journey, Depart on, Arrival on (time 24 Hr clock), Seat no, Name age.

Ans:-

```
#Program
import java.sql.*;

class Mysqlcon {
    public static void main (String args[]) {
        try {
            Class.forName ("com.mysql.jdbc.Driver");
            Connection con = DriverManager.getConnection ("jdbc:mysql://localhost:3306/wpdb", "root", "1234");
            Statement stmt = con.createStatement ();
            stmt.executeUpdate ("insert into Tickets values ('16852677', 'hyd', 'Mumbai', '2020-12-29', '00:50pm', '5:00am', 12, 'Ananya', 20);");
            String str = "insert into Tickets values ('16852699', 'Delhi', 'Kerala', '2021-1-6', '10:00am', '6:50am', 10, 'Ananya', 16);";
            stmt.executeUpdate (str);
            System.out.println ("Two data inserted");
            str = "Select * from Tickets";
            ResultSet rs = stmt.executeQuery (str);
            while (rs.next ()) {
                System.out.println (rs.getString (1) + ";" + rs.getString (2) + ";" + rs.getString (3) + ";" + rs.getString (4) + ";" + rs.getString (5) + ";" + rs.getString (6) + ";" + rs.getString (7) + ";" + rs.getString (8) + ";" + rs.getInt (9));
            }
        } catch (Exception e) {
            System.out.println (e);
        }
    }
}
```

com.close();

}

}

Output:

Two data inserted

16852677, hyd - Mumbai - 2020-12-29 - 00:50pm - 5:00am - 12 - Ananya - 20

16852699 - Delhi - Kerala - 2021-1-6 - 10:00am - 6:50am - 10 - Ananya - 16.