Web Programming Assignment

18W11A1249
G. Akshya

Q1. Write a program to find factorial value of any number entered by the user using javascript.

Ans.

```
<html>
<head>
<script>
function show() {
    var i, no, fact;
    fact = 1;
    no = Number(document.getElementById("num").value);
    for(i=1; i<=no; i+1) {
        fact = fact * i;
    }
    document.getElementById("answer").value = fact;
}
</script>
</head>
<body>
Enter num : <input id="num">
<button onclick="show()">. factorial </button>
<input id="answer">
</body>
</html>
```

output:

Enter num : 5

Factorial 120

Q2. In a company an employee is paid as under:
If his basic salary is less than Rs 1500 then HRA = 10% of basic salary and DA = 90% of salary.
If his salary is either equal to or above Rs 1500, then HRA = Rs 500 & DA = 98% of basic salary. If the employee's salary is input, write the program to find the gross salary,

using javascript event handler.

Ans: Program:

```
<html>
<head>
<script>

function show() {
    var no, ans;
    no = Number(document.getElementById("salary").value);
    if (no<1500) {
        ans = no+(0/100)*no) + ((90/100)*no);
    } else {
        ans = no+(500+((98/100]*no);
    }
    document.getElementById("answer").value = ans;

}
</script>
</head>
<body>
    Enter salary: <input id="salary">
    <button onclick="show()"> gross salary </button>
    <input id="answer">
    </body>

</html>
```

output:

Enter salary | 1203

gross salary | 20383.16

Q3. 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.

```
class solution {
    public static void main (String[] args ) {
        Scanner sc = new Scanner (system.in);
        character c = sc.next();
        if (c=="A" || c=="E" || c=="I" || c=="O" ||
            c=="U") {
                system.out.println ("vowel");
        } else {
            system.out.println ("consonant");
        }
    }
}
```

output:

① Z
consonant

② A
vowel.

Q4. Describe in detail multi-threading concept illustrate with the java code.

Ans. Multi-threading in java 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.

Program:

```
class multiThreadingDemo extends Thread {
        public void run() {
                try {
                    System.out.println("Thread" + Thread.currentThread().
                            getId() + " is running ");
                }
                catch (Exception e) {
                System.out.println("Exception is caught");
                }
        }
}

public class MultiThread {
        public static void main (String[] args) {
                int n = 4;
                for (int i=0; i<n; i++) {
                    multithreadingDemo.Object = new multithreading(i);
                    Object.start();
                }
        }
}
```

Output:

```
Thread 1 is running
Thread 2 is running
Thread 3 is running
```

Q5. Connect to the database using any of the JDBC devices. Insert and display 2 records with bus ticket reservation information.

Ans:

```
import java.sql.*;
public class solution {
```

```
public static void main(String[] args) {
    Connection con = null;
    Statement stmt = null;
try {
    Class.forName("com.mysql.jdbc.Driver");
    con = DriverManager.getConnection("jdbc:mysql://localhost
                    /Bus", "root", "root");
    System.out.println("connection Established");
    stmt = con.createStatement();
String sql = "insert into booking values(130,
        "Hyderabad", "Chennai", "2021-11-12",
    "19:30:10", "2021-01-13", "19:30:10", 4,
    "Person1", 38);
    stmt.executeUpdate(sql);
    sql = "insert into booking values (131, "Hyderabad",
"Bangalore", "2021-01-13", "2021-01-13 19:10:00",
"2021-01-14 08:30:10", 5, "Person2", 40);
    stmt.executeUpdate(sql);
    System.out.println("2 records have been successfully
    executed");
sql = "Select * from booking";
ResultSet rs = stmt.executeQuery(sql);
while (rs.next()) {
    System.out.println("Ticket no:" + rs.getInt("ticketno"));
    System.out.println("source:" + rs.getString("source"));
    System.out.println("Destination:" + rs.getString("destination"));
    System.out.println("DOJ:" + (rs.getDate("doj")).toString());
    System.out.println("Departure:" + rs.getTimestamp("departure").
                                    toString());
```

```
            System.out.println("Arrival:" + (rs.getTimestamp("arrival")).toString());
            System.out.println("seat no: " + rs.getInt("seat no"));
            System.out.println("name: " + rs.getString("name"));
            System.out.println("age: " + rs.getInt("age"));
            System.out.println("\n");
        }
        con.close();
    } catch (exception e) {
            printStackTrace();
    }
  }
}
```

Output: Connection established

2 records have been successfully executed.

Ticket no: 130
source: Hyderabad
Destination: Chennai
DOJ: 2021-01-12
Departure: 2021-01-12 19:30:10
Arrival: 2021-01-13 19:30:10
seat no: 4
Name: Person 1
Age: 38

Ticket no: 131
source: Hyderabad
Destination: Bangalore
DOJ: 2021-01-13
Departure: 2021-01-13 19:10:00
Arrival: 2021-01-14 08:20:10
seat no: 5
Name: Person 2
Age: 40.