HTML Lab Test Answers

1. Write an HTML code to create a web page consisting of your resume information such as, photo, objective, technical skills, educational details, hobbies, contact information using header tags, tables, ordered and unordered list, image, table, link, paragraph tags.

```
<html>
<body>
<h1>John Doe</h1>
<img src="photo.jpg" width="150" height="150">
<h2>Objective</h2>
<p>To work as a software developer in a growth-oriented company.</p>
<h2>Technical Skills</h2>
<ul>
  <li>C++</li>
  <li>Python</li>
  <li>HTML</li>
</ul>
<h2>Educational Details</h2>
<table border="1">
  <tr><th>Degree</th><th>Institute</th><th>Year</th></tr>
  <tr><td>Bachelor of Science</td><td>XYZ University</td><td>2023</td></tr>
  <tr><td>High School</td><td>ABC School</td><td>2019</td></tr>
</table>
<h2>Hobbies</h2>
<ol>
  <li>Reading</li>
  <li>Traveling</li>
  <li>Photography</li>
</ol>
<h2>Contact Information</h2>
<p>Email: <a href="mailto:johndoe@example.com">johndoe@example.com</a></p>
<p>Phone: 123-456-7890</p>
</body>
</html>
```

OUTPUT

John Doe

Objective

To work as a software developer in a growth-oriented company.

Technical Skills

- C++
- Python
- HTML

Educational Details

Degree	Institute	Year
Bachelor of Science	XYZ University	2023
High School	ABC School	2019

Hobbies

1. Reading
2. Traveling
3. Photography

Contact Information

Email: johndoe@example.com

Phone: 123-456-7890

2. Create a web page for college technical events. Generate student registration form for the same inclusive of student name, branch, USN, event they are participating in. Add hyperlinks for home page, event details page, registration page. Add event photos/ college photos, header and footer.

```
<html>
<body>
<h1>College Technical Events</h1>

<p>
<a href="home.html">Home</a> |
<a href="events.html">Event Details</a> |
<a href="register.html">Registration</a>
</p>

<img src="college.jpg" width="300" height="150">
<img src="event1.jpg" width="300" height="150">

<h2>Student Registration Form</h2>
<form>
 Name:<br>
 <input type="text" name="name"><br>
 Branch:<br>
 <input type="text" name="branch"><br>
 USN:<br>
 <input type="text" name="usn"><br>
 Event:<br>
 <select name="event">
  <option>Coding</option>
  <option>Robotics</option>
  <option>Quiz</option>
 </select><br><br>
 <input type="submit" value="Register">
</form>

<p>© 2025 College Name</p>
</body>
</html>
```

OUTPUT

College Technical Events

Home | Event Details | Registration

Student Registration Form

Name:

Branch:

USN:

Event:

Coding

Register

© 2025 College Name

3. Develop and demonstrate the usage of inline, internal and external style sheet using CSS.

```
<html>
<head>
<style>
p.internal {
  color: green;
  font-weight: bold;
}
</style>
</head>
<body>
<p style="color: red;">This is inline styled text.</p>
<p class="internal">This is internal styled text.</p>
<p class="external">This is external styled text.</p>
<link rel="stylesheet" href="styles.css">
</body>
</html>
```

styles.css (external stylesheet):

```
.external {
  color: blue;
  font-style: italic;
}
```

OUTPUT

This is inline styled text.

This is internal styled text.

This is external styled text.

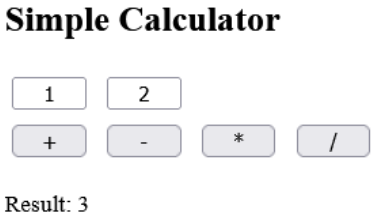
---

4. Write a code to build a simple calculator using HTML and CSS to handle basic arithmetic operations.

```
<html>
<head>
<style>
input {
  width: 50px;
  margin: 5px;
  text-align: center;
}
button {
  width: 50px;
  margin: 5px;
}
</style>
</head>
<body>

<h2>Simple Calculator</h2>
```

```
<input type="text" id="num1">
<input type="text" id="num2"><br>

<button onclick="calculate('+')">+</button>
<button onclick="calculate('-')">-</button>
<button onclick="calculate('*')">*</button>
<button onclick="calculate('/')">/</button>

<p>Result: <span id="result"></span></p>

<script>
function calculate(op) {
  var a = parseFloat(document.getElementById('num1').value);
  var b = parseFloat(document.getElementById('num2').value);
  var res;
  if(op == '+') res = a + b;
  else if(op == '-') res = a - b;
  else if(op == '*') res = a * b;
  else if(op == '/') res = b != 0 ? a / b : 'Error';
  document.getElementById('result').textContent = res;
}
</script>

</body>
</html>
```

OUTPUT



---

5. Write a java script program to convert month number to month name using closures.
a. If the user enters a number less than 1 or greater than 12 or a non-number, have the function write "Bad Number" in the monthName field.
b. If the user enters a decimal between 1 and 12 (inclusive), strip the decimal portion of the no.

```
<html>
<body>

Enter month number: <input id="monthNum" type="text">
<button onclick="showMonth()">Show Month</button>
<p>Month Name: <span id="monthName"></span></p>

<script>
function monthConverter() {
  const months =
["January","February","March","April","May","June","July","August","September","October","Nove
mber","December"];
  return function(num) {
```

```
   num = Math.floor(Number(num));
   if(isNaN(num) || num < 1 || num > 12) return "Bad Number";
   return months[num - 1];
  }
}

const getMonthName = monthConverter();

function showMonth() {
  const input = document.getElementById("monthNum").value;
  document.getElementById("monthName").textContent = getMonthName(input);
}
</script>

</body>
</html>
```

OUTPUT

---

6. Write a java script function named pluralize that:
a. takes 2 arguments, a noun and a number.
b. returns the number and pluralized form, like "5 cats" or "1 dog".
c. Make it handle a few collective nouns like "sheep" and "geese".

```
<html>
<body>
Number: <input id="num" type="text"><br>
Noun: <input id="noun" type="text"><br>
<button onclick="showPlural()">Submit</button>
<p>Result: <span id="result"></span></p>
<script>
function pluralize(noun, number) {
  const specials = {sheep: "sheep", goose: "geese"};
  if(number === 1) return number + " " + noun;
  if(specials[noun.toLowerCase()]) return number + " " + specials[noun.toLowerCase()];
  if(noun.endsWith("y")) return number + " " + noun.slice(0, -1) + "ies";
  return number + " " + noun + "s";
}
function showPlural() {
  const n = Number(document.getElementById("num").value);
  const word = document.getElementById("noun").value.trim();
  if(isNaN(n) || word === "") {
    document.getElementById("result").textContent = "Enter valid number and noun";
    return;
  }
  document.getElementById("result").textContent = pluralize(word, n);
}
</script>
```

```
</body>
</html>
```

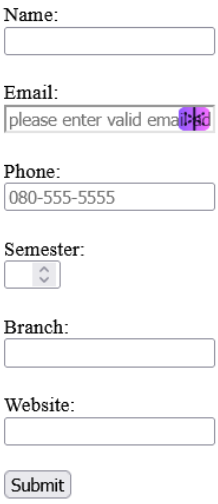
OUTPUT

---

7. Design a Student Form using HTML5 which has following fields:
a) Name: Required must be characters
b) Email: Validation placeholder: please enter valid email address
c) Phone: accept numbers in the following format (080-555-5555)
d) Semester: For the range 1 to 8
e) Branch: Data list
f) Website: Required pattern of the form-http://

```
<html>
<body>
<form>
  Name:<br>
  <input type="text" name="name" pattern="[A-Za-z\s]+" required><br><br>

  Email:<br>
  <input type="email" name="email" placeholder="please enter valid email address"
required><br><br>

  Phone:<br>
  <input type="tel" name="phone" pattern="\d{3}-\d{3}-\d{4}" placeholder="080-555-
5555"><br><br>

  Semester:<br>
  <input type="number" name="semester" min="1" max="8"><br><br>

  Branch:<br>
  <input list="branches" name="branch">
  <datalist id="branches">
    <option value="CSE">
    <option value="ECE">
    <option value="ME">
    <option value="CE">
  </datalist><br><br>

  Website:<br>
  <input type="url" name="website" pattern="http://.*" required><br><br>

  <input type="submit" value="Submit">
</form>
</body>
</html>
```

OUTPUT:

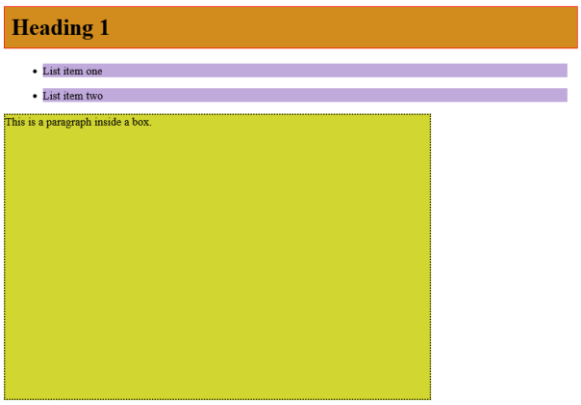


---

8. Create a web page with the following characteristics using BOX Model
a. h1's have 1px red solid borders, background color #D18C1D, and 10px of space between the content and the border (padding)
b. List items have 15px extra space around them (margin) and background color #C0A9DB
c. Paragraphs are contained in 600px by 400px boxes with 2px black dotted borders and background color #D1D631

```
<html>
<head>
<style>
h1 {
  border: 1px solid red;
  background-color: #D18C1D;
  padding: 10px;
}
li {
  margin: 15px;
  background-color: #C0A9DB;
}
p {
  width: 600px;
  height: 400px;
  border: 2px dotted black;
  background-color: #D1D631;
}
</style>
</head>
<body>
<h1>Heading 1</h1>
<ul>
  <li>List item one</li>
  <li>List item two</li>
</ul>
<p>This is a paragraph inside a box.</p>
</body>
</html>
```

OUTPUT



---

9. Write a java script program to implement Stack and Queue using modules.

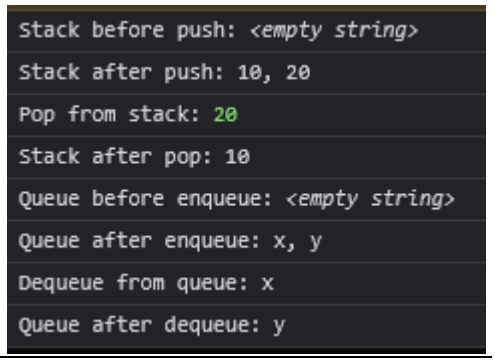
module.js file contents

```
export class Stack {
  constructor() { this.items = []; }
  push(e) { this.items.push(e); }
  pop() { return this.items.pop(); }
  print() { return this.items.join(", "); }
}
export class Queue {
  constructor() { this.items = []; }
  enqueue(e) { this.items.push(e); }
  dequeue() { return this.items.shift(); }
  print() { return this.items.join(", "); }
}
```

Index.html contents

```
<html>
<body>
<script type="module">
import { Stack, Queue } from './module.js';
const s = new Stack();
console.log("Stack before push:", s.print());
s.push(10);
s.push(20);
console.log("Stack after push:", s.print());
console.log("Pop from stack:", s.pop());
console.log("Stack after pop:", s.print());
const q = new Queue();
console.log("Queue before enqueue:", q.print());
q.enqueue('x');
q.enqueue('y');
console.log("Queue after enqueue:", q.print());
console.log("Dequeue from queue:", q.dequeue());
console.log("Queue after dequeue:", q.print());
</script>
</body>
</html>
```

OUTPUT (in console)
Stack before push:
Stack after push: 10, 20
Pop from stack: 20
Stack after pop: 10
Queue before enqueue:
Queue after enqueue: x, y
Dequeue from queue: x
Queue after dequeue: y



---

10. Using node.js Express and Mongo to implement 'FinalYears' database which accepts 'USN', 'Name' and 'Company_name' (by campus selection) as fields and store it in a database. Display the list of students who are selected for 'Infosys'

Step 1: Setup Express server and MongoDB connection

```
// server.js
const express = require('express');
const mongoose = require('mongoose');
const app = express();

app.use(express.json());

mongoose.connect('mongodb://localhost:27017/FinalYears', {
  useNewUrlParser: true,
  useUnifiedTopology: true,
});

const studentSchema = new mongoose.Schema({
  USN: String,
  Name: String,
  Company_name: String,
});

const Student = mongoose.model('Student', studentSchema);

// POST student data
app.post('/addStudent', async (req, res) => {
  const { USN, Name, Company_name } = req.body;
  const student = new Student({ USN, Name, Company_name });
  await student.save();
  res.send('Student added');
});

// GET students selected for Infosys
app.get('/infosysStudents', async (req, res) => {
  const students = await Student.find({ Company_name: 'Infosys' });
  res.json(students);
});

app.listen(3000, () => console.log('Server started on port 3000'));
```

EXPECTED OUTPUT:
POST /addStudent with JSON:
{ "USN": "1MS20CS001", "Name": "Alice", "Company_name": "Infosys" }

Response:
Student added

GET /infosysStudents response JSON:
[
{ "_id": "...", "USN": "1MS20CS001", "Name": "Alice", "Company_name": "Infosys", "__v": 0 },
...
]

How to test:

1. Run npm init -y and install dependencies:
   npm install express mongoose
2. Run MongoDB locally.
3. Run this server with node server.js.
4. Use Postman or curl to POST student data and GET Infosys students.

The above question is nonsense she expects us to understand and execute on a whim under a week
EOF