

MCSL-216 LAB QUESTIONS

Course Code : MCSL-216

Course Title : DAA and Web Design Lab

This Record has two sections as per Lab Manual . Your Lab Records will carry 40 Marks (20 Marks for each section). Rest 20 marks are for viva voce. You may use illustrations and diagrams to enhance the explanations. Please go through the guidelines regarding Lab records given in the programme guide for the format of presentation.

SECTION-1

DESIGN AND ANALYSIS OF ALGORITHMS LAB

>>Session 1: Implementation of Simple Algorithms

Q1. Implement Euclid's algorithm to find GCD (15265, 15) and calculate the number of times mod and assignment operations will be required.

Q2(i). Implement Horner's rule for evaluating polynomial $P(x) = 6x^6 + 5x^5 + 4x^4 - 3x^3 + 2x^2 + 8x - 7$ at $x = 3$. Calculate how many times **(i)** multiplications and addition operations will take **(ii)** how many times the loop will iterate .

Q2(ii) .Apply a brute force method to implement the above polynomial expression(Q2(i)) and compare it with Horner's method in terms of number of multiplication operations

Q3. Implement multiplication of two matrices $A[4,4]$ and $B[4,4]$ and calculate **(i)** how many times the innermost and the outermost loops will run **(ii)** total number of multiplications and additions in computing the multiplication

Q4. Implement left to right and right to left binary exponentiation methods for the following problems: **(i)** 4^{512} **(ii)** 3^{31} Calculate the followings for problems **(i)** and **(ii)** How many times the loops will be executed? How many times the multiplication and additions operations will be executed?

Q5. Implement Bubble Sort algorithm for the following list of numbers: 55 25 15 40 60 35 17 65 75 10 Calculate **(i)** a number of exchange operations **(ii)** a number of times comparison operations **(iii)** a number of times the inner and outer loops will iterate?

Q6. Implement Selection Sort algorithm on a similar set of data as in Q5 and **(i)**perform similar operations on the above example **(ii)** make a comparison between the two algorithms in terms of best case and worst case complexities .

>>Session 2: Fractional Knapsack Problem

Q1.(P1, P2, P3, P4, P5, P6, P7) = (15, 5, 20, 8, 7, 20, 6)

(W1, W2, W3, W4, W5, W6, W7) = (3, 4, 6, 8, 2, 2, 3)

Maximum Knapsack Capacity = 18

Q2.(P1, P2, P3, P4, P5) = (20, 30, 40, 32, 55)

(W1, W2, W3, W4, W5) = (5, 8, 10, 12, 15)

Maximum Knapsack Capacity = 20

Q3.(P1, P2, P3, P4, P5, P6, P7) = (12, 10, 8, 11, 14, 7, 9)

(W1, W2, W3, W4, W5, W6, W7) = (4, 6, 5, 7, 3, 1, 6)

>>Session 3 : Task Scheduling Algorithm

Q1. Apply a brute force approach to schedule three jobs J1, J2 and J3 with service times as 5, 8, 12 respectively. The actual service time units are not relevant to the problems. Make all possible job schedules , calculate the total times spent in jobs by the system. Find the optimal schedule (total time). If there are N jobs , what would be the total number of job schedules?

Q2. Implement the task scheduling algorithm on your system to minimize the total amount of time spent in the system for the following problem:

Job	Service Time
1	5
2	10
3	7
4	8

Q3. Consider the following jobs, deadlines and profits. Implement the task scheduling algorithm with deadlines to maximize the total profits.

Jobs	Deadlines	Profits
1	3	50
2	4	20
3	5	70
4	3	15
5	2	10
6	1	47
7	1	60

>>Session 4: Huffman's Coding Algorithm

Q1. Apply Huffman's algorithm to construct an optimal binary prefix code for the letters and its frequencies in the following table.

Letters	A	B	I	M	S	X	Z
Frequency	10	7	15	8	10	5	2

Show the complete steps.

Q2. Implement Huffman's coding algorithm and run on the problem instance of Q1.

Q3. What is an optimal binary tree and Huffman code for the following set of frequencies? Find out an average number of bits required per character. Show the complete steps.

A:15 B:25 C:5 D:7 E:10 F:13 G:9

Q4. A file contains only colon, spaces, new line character and digits in the following frequency: colon (80), space (500), new line (110), commas (500), 0 (300), 1 (200), 2 (150), 3(60), 4 (180), 5 (240), 6 (170), 7 (200), 8 (202). Using the Huffman's algorithm to construct an optimal binary prefix code.

>>Session 5: Divide and Conquer Technique

Q1. Implement a recursive binary search algorithm on your system to search for a number 100 in the following array of integers. Show the processes step by step:

10 35 40 45 50 55 60 65 70 100

Draw recursive calls to be made in this problem.

Q2. Suppose that we are required to search among 512 million items in a list using binary search algorithm. What is the maximum number of searches needed to find a given item in the list or coming to the conclusion that the item is not found in the list.

Q3. Implement Merge Sort algorithm to sort the following list show the process step by step.

200 150 50 100 75 25 10 5

Draw a tree of recursive calls in this problem.

Q4. Implement Quick Sort's algorithm on your machine to do sorting of the following list of elements

12 20 22 16 25 18 8 10 6 15

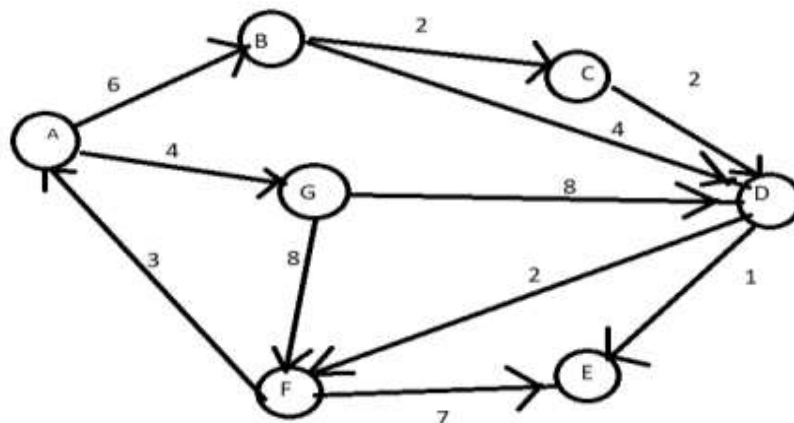
Show step by step processes.

Q5. Examine the performance of Quick Sort algorithm implemented in the previous problem for the following list in terms of a number of comparisons, exchange operations and the number of times the loop will iterate?

6 8 10 12 15 16 18 20 22 25

Q6. Implement the Stassen's multiplication algorithm two matrices A and B of $n \times n$, where n is a power of 2 on different problem instances and compare it(all the instances) in terms of a number of multiplications and additions required.

>>Session 6: Single Source Shortest Path Algorithm



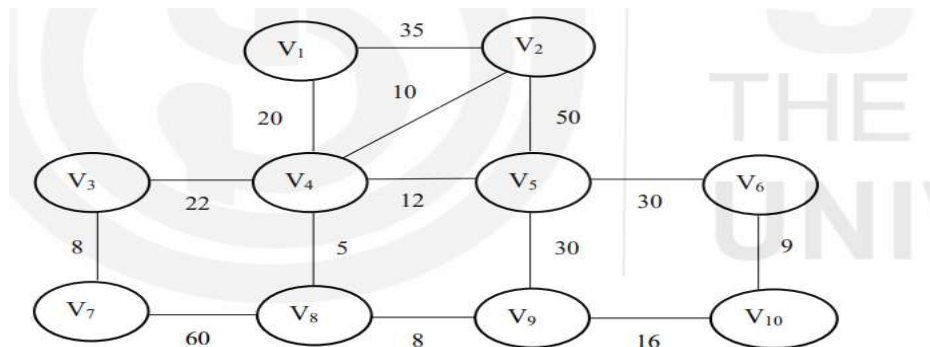
Q1. Find the shortest path from A to the rest of vertices .

Q2. Find the shortest path from B to the rest of nodes .

Q3. Find the shortest path from A to the rest of vertices but there are negative weights(-8) between G and F and (-3) between F and A .

>>Session 7: Minimum Cost Spanning Tree

Q1. Implement Prim's algorithm to find a minimum cost spanning tree (MCST) in the following graph. Show all the processes.



Q2. Implement Kruskal's algorithm to find a MCST for the following graph. Show all the processes.

Q3. Analyze the performance of both the algorithms on different problem instances and write a brief report .

>>Session 8: Implementation of Binomial Coefficient Algorithm

Q1. Implement a binomial coefficient problem using Divide and Conquer technique

Q2. Implement a binomial coefficient problem using Dynamic Programming technique .

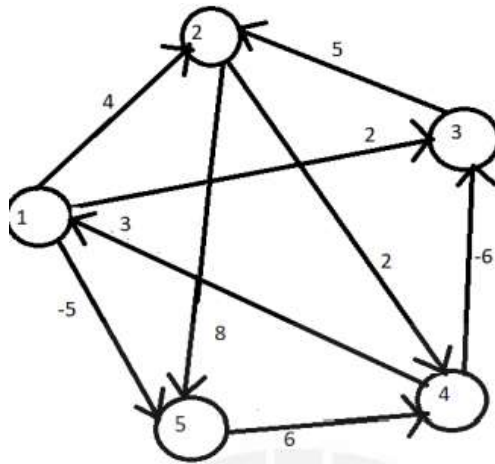
Q3. Study the performance of both implementations using five problem instances in terms of the efficiency of both the approaches for large and small values of n and k in the problem instances.

>>Session 9: Floyd and Warshall's Algorithm for All Pair Shortest Path Algorithms

Q1. Apply the Floyd and Warshall's algorithm for the following graph. Show the matrix D_5 of the graph and find out the shortest path .

Q2. Apply the Floyd and Warshall's algorithm to compute the shortest path to the following graph. Compute D_5 matrix of the graph .

Q3. Implement the all pair shortest path algorithms using different graphs.



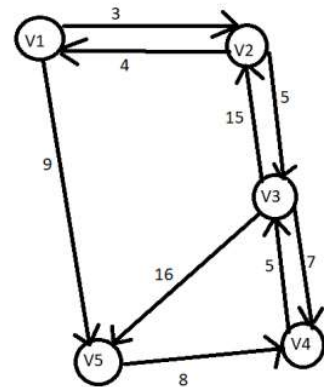
>>Session 10: Chained Matrix Multiplication

Q1. List different orders for evaluating the product of A,B,C,D,E matrices.

Q2. Find the optimal order for evaluating the product

$A * B * C * D * E$ where :

- A is (10 * 4)
- B is (4 * 5)
- C is (5 * 20)
- D is (20 * 2)
- E is (2* 50)



Q3. Implement the chained matrix multiplication and print the optimal parentheses algorithms and study the performance of the algorithms on different problem instances.

>>Session 11: Optimal Binary Search Tree

Determine the cost and structure of an optimal binary search tree for a set of $n=7$ keys with the following properties. Show the step by step processes.

Q1.

i	0	1	2	3	4	5	6	7
p_i		0.04	0.06	0.08	0.02	0.10	0.12	0.14
q_i	0.06	0.06	0.06	0.06	0.05	0.05	0.05	0.05

Q2.

i	0	1	2	3	4	5
p_i		0.15	0.10	0.05	0.10	0.20
q_i	0.05	0.10	0.05	0.05	0.05	0.10

Q3. Implement the optimal binary search tree algorithm on your system and study the performance of the algorithm on different problem instances.

SECTION-2

WEB DESIGN LAB

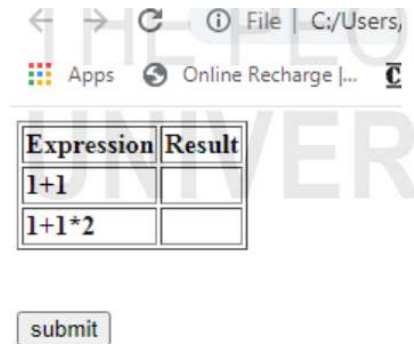
>>Session 1: Preliminaries

1.5 Exercises

1. Write a JavaScript code to perform the two-digit arithmetic operations: Addition, Subtraction, Multiplication and Division.
2. Write a JavaScript code to print 1 for 1 time, 2 for 2 times, 3 for 3 times and so on upto 10.
3. Design a web page that describes you. Your page must include following details (extra details and creativity are welcome):
 - Your personal details, i.e. name, gender etc .
 - 2-3 lines describing yourself as About Me. Make the most important word bold to emphasize them.
 - Details of classes in tabular form you are taking right now (as per your time table).
 - Details of your favourite movies, books, or TV shows, in reverse order(using CSS), list atleast 3 from each. You can use images, hyperlinks, colourful text to make this section attractive.
 - Include a section showing your interest and also not of your interest jobs. In this section also you can use images, showing you're happy in doing your interest jobs and the other to represent you when you're sad.
 - In this section show something interesting about one or more people of your neighbours.

>>Session 2: HTML Forms

1. Create an HTML webpage, as shown in figure. The user can specify any expression as input. Use form element to collect the user data.



Expression	Result
1+1	
1+1*2	

submit

2. Create a scrolling list that shows five items from the following list of PC processors: Intel i7, Motorola 68000, AMD Athlon , Intel 8088, AMD Ryzon, Intel Pentium MMX, Intel i3, Intel Pentium II, Intel Pentium III, Intel i5, Intel Celeron, PowerPC G3, PowerPC G4,.

3. Design a webpage for a software company to accept on-line job applications. The webpage must be designed to use form with elements:

- Position applied for (autofocus), name, nationality, date of birth (selected from an auto picker), address (in a text area), telephone number and email (required).
- Educational history and qualifications.
- Work experience/employment/training in terms of employer history and number of years of experience selected from a slider. Set maximum years of experience to 10 years.
- Personal statement.
- Two referees including names, occupation, relationship, address, telephone.

>>Session 3: Cascading Style Sheet (CSS)

1. Create an HTML website to display the time table of all the 4 years of B. Tech. (even semester of each year). It should contain a home page where link to each year's time table is listed. On click to any year time table a new web page should be open displaying the corresponding time table. Apply the following effects on the table using CSS:

- Common to all:

- Display day names (Mon, Tue etc...) in bold format with the first letter in the dayname in uppercase.
- Display lunch slightly in bigger font other than the remaining text

-Specific to each year:

- For 1st year

*Apply blue as the background colour and white for the colour of the text in the table header.

- For 2nd year

*Apply green as the background colour and white for the colour of the text in the table header.

- For 3rd year

*Apply purple as the background colour and white for the colour of the text in the table header.

- For 4th year

*Apply yellow as the background colour and black for the colour of the text in the table header.

2. Create an HTML webpage to show personal information i.e. name, class, qualifications, photo , address etc. Make use of tables as when possible. Apply the following style information using CSS:

- Display the heading of the page in Arial font and with font size of 18px.
- Align all the field names like Name, Class, Photo etc to left in the table.
- Apply yellow color as background colour in left side cells contains field names like Name, Class etc...
- Set college logo as background image to the web page.

>>Session 4: JavaScript HTML DOM

1. Design an HTML page using JavaScript as shown below. When Try button is pressed, the Full Name should be shown below the Try button. (Hint: Use concatenation to get Full Name.

Accessing all the elements of HTML Form with document.forms

First name:
Last name:

Click on "Try" button to show the value of elements in the form.

On Click on Try:

Accessing all the elements of HTML Form with document.forms

First name:
Last name:

Click on "Try" button to show the value of elements in the form.

Bhagat
Singh

2. Modify the below snippet of the code to change the src attribute of the image element to “pqr.pg” using HTML DOM.

```

```

```
<script>
```

```
//write your code here.
```

```
</script>
```

3. Write the JavaScript code to change the color of the HTML element h2 to green, on click the “change color” button.

>>Session 5: JavaScript HTML DOM events and Event Listener

1. Design an HTML page as shown below. Implement the event “On button Click display the date” using HTML DOM events. Before click on submit button:

Click on submit button to display Date and Time.

After click on submit button:

Click on submit button to display Date and Time.

Mon May 31 2021 13:04:14 GMT+0530 (India Standard Time)

2. Design an HTML page to show that ,when this page is loaded it must check the cookies status of user's browser and display a message accordingly in the popup box i.e if cookies are enabled show : "Cookies are enabled" and if not then show: "Cookies are disabled". Implement the event "On load" using HTML DOM events.
3. Whenever user input a text in the text box in any case (i.e. upper or lower case), automatically it should be converted to uppercase. Design an HTML page to implement above functionality using HTML DOM event "On Change".
4. Design HTML page to implement HTML DOM events by changing an image on the following events: onmouseover, onmouseout, onmousedown, and onmouseup Events.

>>Session 6: HTML DOM Navigations

1. Which function on the document object should be used to retrieve a HTML element uniquely?
2. Which JavaScript function can be used to dynamically add a event listener to a HTML Element?
3. Which method should be used to add a new HTML child element?
4. Using the DOM API, create a product details page which shows the prices of a laptop and a cell phone. You may paragraph tag to wrap these elements. When the user clicks on add button, you should able to add TV with price details and when you click on delete button then cell phone details will be deleted.



>>Session 7: JavaScript Error Handling

1. Write an HTML code using JavaScript to add two numbers taken as input from user and display the result of division operation on them i.e. a/b. If an error occurs, show the error. Use error handling concepts in JavaScript.

2. A teacher has created a `gradeLabs` function that verifies if student programming labs work. This function loops over an array of JavaScript objects that should contain a `student` property and `runLab` property. The `runLab` property is expected to be a function containing the student's code. The `runLab` function is called and the result is compared to the expected result. If the result and expected result don't match, then the lab is considered a failure.

3. There is an array of animals. The user is asked to enter the index for the animal they want to see.

i) If the user enters an index that does NOT contain an animal, the code will throw a `TypeError` when name is referenced on an undefined value.

ii) Update the above program to print out the index the user entered. We want this message to be printed EVERY time the code runs .

4. Assume you have a function `primitiveMultiply` that in 20 percent of cases multiplies two numbers and in the other 80 percent of cases raises an exception of type `MultiplicatorUnitFailure`. Write a function that wraps this clunky function and just keeps trying until a call succeeds, after which it returns the result. Make sure you handle only the exceptions you are trying to handle.

>>Session 8: JavaScript Classes

1. Let us assume you need to organize a library of some electronic manuals and novels for a particular company. For each book, you need to store following information:

- The title
- The author
- The copyright date
- The ISBN
- The number of pages
- The number of times the book has been checked out.
- Whether the book has been discarded .

Company wants to perform certain actions when the any book is outdated. The manual books must be exited after 5 years while a novel book should be exited if its checked out time crossed 100.

Task 1. Construct three classes that hold the information needed by company. One class should be a **Book** class and two child classes of the Book class called **Manual** and **Novel**. Each class will contain two methods. One will be a constructor. The other one will either be in charge of disposal of the book or updating the property related to the number of times a book has been checked out.

Task 2: Create an object of the Novel class and Manual class with some valid details.

Task 3: Write the method to count the duration of a manual book and checkout time for novel book so that they can be discarded based on the duration.

2. Create a object 'product' with properties :ID ,Name, Description and Price .Then create function for this object called displayProductDetails(). This function when invoked should display the name, description and discounted price of the product. For the calculation of discounted price you need to create another function CalculateDisc(percentage). The discount percentage would be given by user.

>>Session 9: JavaScript Asynchronous programming

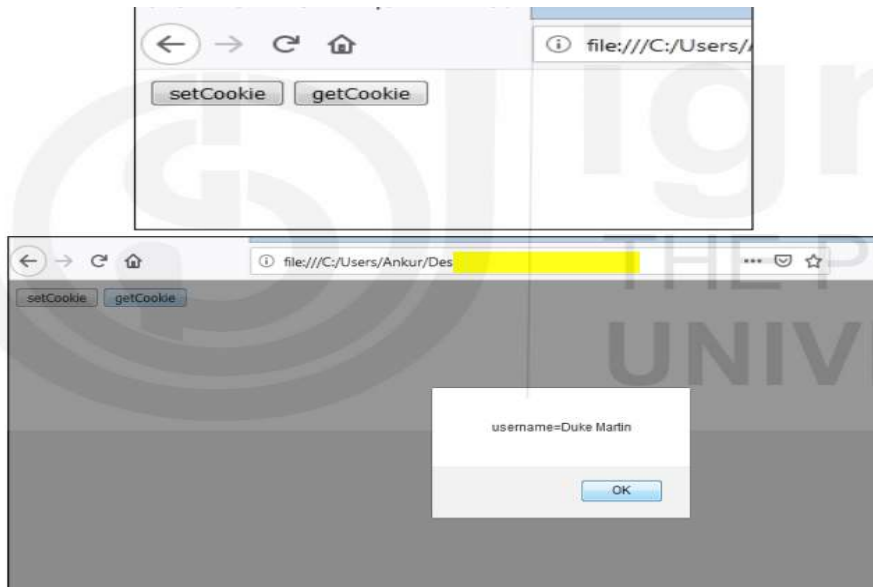
1. Implement a clock in HTML which shows the live time in the format HH:MM:SS. Use the JavaScript setInterval() method and a callback function by displaying the system time every second.

2. Sometimes we use external resource files in our HTML page. The content of an external file cannot be used until loaded completely. This can be implemented with JavaScript Callback. Design a web page which loads an external HTML file on the event: onLoad. Hint: use the inbuilt class XMLHttpRequest() to use its functions: open(), onLoad() etc. to load an external HTML file.

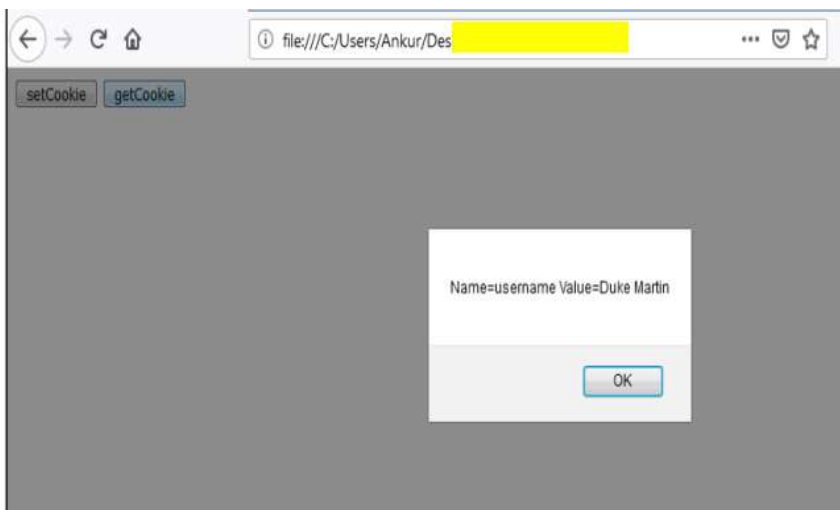
3.The village crows own an old scalpel that they occasionally use on special missions²say, to cut through screen doors or packaging. To be able to quickly track it down, every time the scalpel is moved to another nest, an entry is added to the storage of both the nest that had it and the nest that took it, under the name "scalpel", with its new location as the value.This means that finding the scalpel is a matter of following the breadcrumb trail of storage entries, until you find a nest where that points at the nest itself. Write an async function locateScalpel that does this, starting at the nest on which it runs. You can use the anyStorage function defined earlier to access storage in arbitrary nests. The scalpel has been going around long enough that you may assume that every nest has a "scalpel" entry in its data storage.

>>Session 10: JavaScript Cookies

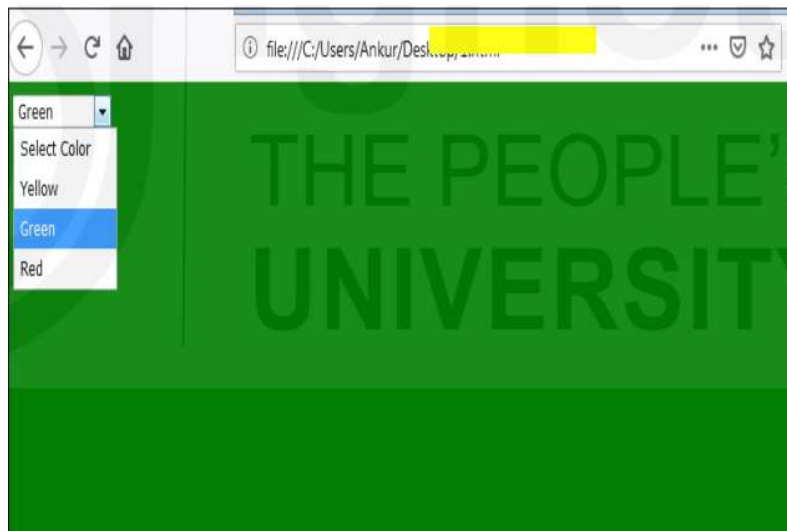
1. Create an HTML web page, as shown below. The cookie will be set on pressing setCookie button and the stored cookie value will be displayed on pressing getCookie button.



2. Considering exercise 1, modify the web page to display the cookie's name-value pair separately as shown in figure below .



3. A list of color is provided in the form of drop down list. The background color of the web page changes as per the selection of the color from the list by the user. At the same time the chosen color is passed to the cookie. The cookie stores the last choice of a user in a browser which will be used on reloading the web page to display the same color as background color.



4. Create an HTML web page, as shown below. The cookie1 and cookie2 will be set on pressing Set Cookie1 or Set Cookie2 button and the stored cookie value will be displayed on pressing Get Cookie1 or Get Cookie2 button respectively. On the other hand selectively cookie can be deleted by pressing Delete Cookie1 or Delete Cookie2 button. Display all cookies button will show all the stored cookies.

