THINGS TO KNOW:

1. **Lab report must contain following sections: (order must be maintained)**
   a) **Title /Question**
   b) **Theory: The brief overview of the concept /techniques/syntax/technology used in the program**
   c) **Code: The complete code (markups, styles and scripts)**
   d) **Output: Screenshot of the output (in console and/or browser window)**
2. **Output screen should be captured (use snipping tool), printed and attached in the report. Other contents must be handwritten.**
3. **Every source code must include the printing statements to print following information after your main output:**

   Lab No.:

   Name:

   Roll No./Section :

4. **Contents should be written on single side of A4 sized paper.**
5. **The works must be submitted within specified deadline.**
6. **Cover page and index page should be attached in the report appropriately.**

<u>Index Page Format (can be printed)</u>

List of Lab Works

Lab No.	**Title /Question**	**Submission Date**	**Signature**	**Remarks**
1(a)	This is sample	2079/11/12		
1(b)	This is also sample	2078/11/12		

Scripting Language / BCA – 4th Semester Lab Works (Part-1)

1. Write required markups and scripts for performing following tasks
    a. Display "Hello World" on <h2> tag using JavaScript inside <script> tag.(Hint: Use *document.getElementById('idName').innerHTML*)
    b. Create a JavaScript file to demonstrate the difference between **let** and **var** keywords and link that file to HTML file using <script> tag.
    c. WAP to display type of different variables using *typeof* operator
    d. WAP to show the difference between == and === operators
2. Learning conditional statements in JS.
    a. Create a web page that asks user to enter a number form prompt dialog box and displays whether the number is even or odd in alert dialog box using *if-else* statements.
    b. Create a web page that asks user to enter a number form prompt dialog and displays whether the number is single digit number, double digit number, triple digit number or multi digit number using *if-else-if* statements.
    c. Use prompt dialogs to take two numbers and an operator (+,-,*, /, %) from user and display the result on alert box according to the operator entered using *switch-case* statements.
3. Understanding basic loops in JS.
    a. Write a script to take a number form user using prompt dialog and display the multiplication table of that number on web page using *for-loop*.
    b. Create a web page to take a number from prompt dialog and display the sum of digits of that number in an alert box using *while-loop. (if number is 1457 the result should be (1+4+5+7) = 12).*
    c. Use prompt dialogs to take numbers and a confirm dialogs to ask whether to take another number or not. Write a script that takes numbers from user until user want to enter another number and display the sum of the all entered numbers using *do-while -loop.*
4. Working with arrays in JS.
    a. Write a program to demonstrate *various ways to create an array* in JS.

b. Write a program to demonstrate *two-dimensional* and *three-dimensional array creation* and *accessing elements* from those arrays.

c. Write a program to illustrate *forEach()* method to iterate an array.

d. Write a program to illustrate *for-of* and *for-in* statements in JS.

e. Write a program to *delete an element* from given position of an array.

f. Write a program to illustrate use of following *array methods*:

- includes()
- find()
- findIndex()
- indexOf()
- lastIndexOf()
- join()
- concat()
- pop()
- push()
- reverse()
- shift()
- unshift()
- slice()
- map()
- sort()
- every()

5. JS Functions

a. Write a program to illustrate how a *function is created* using

- *function declaration* statement and
- *function expression*

b. Write a program to demonstrate *anonymous function* and *arrow functions* in JS

c. Write a program to demonstrate following *function invocation techniques in JS*:

- As functions
- As methods

- As constructors
- Indirectly through their call() and apply() methods
- Implicit function invocation

d. Write a program to prove the statement "*arguments in JavaScript are passed to function by value*".

e. Write a program to demonstrate the use *default parameters* in JS function.

f. Write a program to illustrate the use of *arguments object* in JS function.

g. Write a program to illustrate the use of *rest parameters* in JS function.

6. Learning basic ideas about JS object.

a. Write a program to demonstrate how an *object is created using object literal* and how the *properties of the objects are accessed*.

b. Write a program to demonstrate JS *object creation using Object.create()* function.

c. Write a program having a *constructor function* and *create some objects* from that function.

d. WAP to demonstrate *deletion of a property* from JS object.

e. WAP having JS *object with function, another object and array as values of it's properties*. Program should contain statements to access those properties and display them.

f. WAP to demonstrate *getters* and *setters* in JS object.

7. Exploring some *built-in JS objects*.

Explore the following built-in JS objects and write programs to show the use of some properties and methods of these objects. (Explore all or at least 5 important properties and methods)

(Hint : To work with Math object, consider the properties like E, PI, LN2, LN10, SQRT2 i.e. Math.E, Math.PI, Math.LN2,Math.LN10, Math.SQRT2 and methods like Math.floor(), Math.ceil(), Math.sqrt(), Math.round(), Math.pow())

- Math
- String
- Number
- Object
- Function

- Error
- Boolean
- Array

8. Working with Date object.
   a. Write a program to demonstrate *different ways to crate date object* in JS
   b. Write a program to show the use of following *getter methods of date object* in JS

      getDate()

      getDay()

      getFullYear()

      getHours()

      getMonth()

      getMilliSeconds()

      getSeconds()

      getMinutes()

      getTimeZoneOffset()

      getYear()

      getUTCFullYear()

      getUTCDay()

      getUTCDate()

      getUTCMonth()

   c. Write a program to show the use of following *setter methods of date object* in JS

      setDate()

      setDay()

      setFullYear()

setHours()

setMonth()

setMilliSeconds()

setSeconds()

setMinutes()

setYear()

setUTCFullYear()

setUTCDay()

setUTCDate()

setUTCMonth()

d. Create a web page that *takes date of birth as input and display the age*
    - In years
    - In years, months and days

e. Create a web page having a *digital clock blinking each second* and displaying correct time. The page also should display the current date.

f. Write a program to *display the date before after given*
    - Days
    - Hours

9. Create a web page containing *stop watch* (must contain start, pause and reset buttons)

10. Event Handling in JS

a. WAP to demonstrate *different ways of registering events* in JS.(Hint: use click event for each technique)

b. Create two text fields to take two numbers and three buttons each labeled with 'add', 'subtract' and 'multiply'. Write the scripts such that *when a button is clicked corresponding result will be displayed on the alert dialog box*.

c. WAP to demonstrate following *mouse event* handling
    - Click

- Mouseover
- Mouseout
- Mousedown
- Mouseup
- Mousemove

e) WAP to demonstrate following *keyboard event* handling

- Keydown
- keyup

f) WAP to demonstrate following *form event* handling

- Focus
- Submit
- Blur
- Change

g) WAP to demonstrate following *window/document* event handling

- Load
- Unload
- Resize

h) WAP to demonstrate *input event* handling.

11. Interacting with browser using Browser Object Model (BOM)

a. Write a program to display *inner and outer width and heights* of browser window on an alert box when a button is clicked.

b. Write a program to demonstrate the use of *open()* and *close()* window methods for opening and closing the pop-up windows. (use two buttons for opening and closing)

c. WAP to illustrate the use of *setTimeOut()* and *clearTimeOut()* methods of window objects.

d. WAP to illustrate the use of *setInterval()* and *clearInterval()* methods of window objects.

e. WAP to illustrate the use of different properties of *location* object.

f. WAP to illustrate the use of following methods of *location* object.

assign() , replace() , reload()

g. Create a web page and perform activities to illustrate the following properties and methods of *history* object

   - length
   - back()
   - forward()
   - go()

h. WAP to illustrate the usages of different properties and methods of *navigator* object.

i. Create a web page to demonstrate the use of *frame* object with different properties and methods.