Q1. What is the difference between Cookie / Local Storage / Session Storage?

**Cookie**: Cookies are a small amount of data stored on both client-side and server-side. Cookies only allow 4KB of data to be stored.

**Local Storage**: Local storage is a type of web storage which stores data on client-side. It allows to save key/value pairs to store in a web browser with no expiration date. The data will be clear from browser once user clear the cache. Local storage allows to store nearly 10MB of data.

**Session Storage**: Session storage is also a type of web storage which will expire once the session ends , user clicks on next page of the website or closes the tab or browser window. It also allows to store nearly 10MB of data

Q2. What is the significance of, and reason for, wrapping the entire content of a JavaScript source file in a function block?

**Ans**. Main purpose is to avoid polluting global scope and to avoid overriding already existing variables

Q3. What will the code below output? Explain your answer.

console.log(0.1 + 0.2);

**Output**: 0.30000000000000004

// The output shows the sum of two numbers which are added using addition operator and the value in the output is also of number type

console.log(0.1 + 0.2 == 0.3);

**Output**: false

//The output here shows the sum of two numbers which is then compared to the result value given using “equal to” comparison operator which will give output as a Boolean value which is “false” in this case.

Q4. Write a sum method which will work properly when invoked using either syntax below.

console.log(sum(2,3));   // Outputs 5

console.log(sum(2)(3));  // Outputs 5

function sum(a,b){

return (a+b); //or return parseFloat(a) +parseFloat(b); in case value passed is a decimal number.

}

console.log(sum(2,3)); // 5

Q5. What is the output of the following code?

      var length = 10;

function fn()  {

console.log(this.length);

}

var obj = {

length: 5,

method: function(fn) {

fn();

arguments[0]();

}

};

obj.method(fn, 1);

**Output**:

10

2