**Practical-1**

**Aim:-**

Create a program that declares and initializes variables of different data types

(string, number, boolean) and displays their values. Write a function that takes

two numbers as parameters and returns their sum

**Code:-**

function addNum(n1, n2) {

    return n1 + n2;

  }

  let Name = "Shivam Ardeshna";

  let age = 121;

  let bool = true;

  let float = 22.57;

  console.log("Name:", Name);

  console.log("Age:", age);

  console.log("Bool:", bool);

  console.log("float:", float);

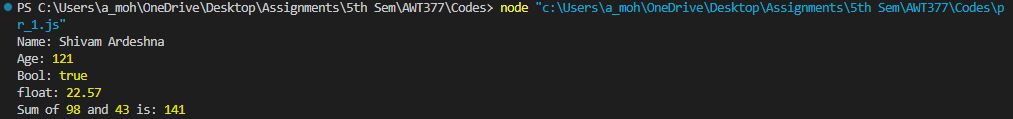
  let n1 = 98;

  let n2 = 43;

  let ans = addNum(n1, n2);

  console.log("Sum of", n1, "and", n2, "is:", ans);

**Output:-**

****

**Practical-2**

**Aim:-**

Create an array of numbers and perform the following operations:

=> Find the length of the array.

=> Access and display specific elements using indexing.

=>Use array methods like push() ,pop(), shift(), unshift(), join(),

delete(),concate(),flat(),splice() and slice() to modify the array.

Create an object representing a person with properties like name, age, and

gender. Implement a function that displays the person's details.

**Code:-**

let numArrays = ["Shivam", 18, true, 11.9];

console.log("Length of the array:", numArrays.length);

console.log("Element at index 2:", numArrays[2]);

numArrays.push({ 1: 1 });

console.log("Array after push:", numArrays);

numArrays.pop();

console.log("Array after pop:", numArrays);

numArrays.shift();

console.log("Array after shift:", numArrays);

numArrays.unshift(0);

console.log("Array after unshift:", numArrays);

console.log("Display array using join():", numArrays.join(", "));

delete numArrays[2];

console.log("Array after delete:", numArrays);

let anotherarray = [1, 2, 3];

let concatarr = numArrays.concat(anotherarray);

console.log("Concatenated array:", concatarr);

let nestedArray = [1, [2, 3], [4, [5, 6]]];

let flatArray = nestedArray.flat(3);

console.log("Flattened array:", flatArray);

numArrays.splice(1, 2, 5, 8);

console.log("Array after splice:", numArrays);

let slicearr = numArrays.slice(1, 3);

console.log("Sliced array:", slicearr);

let Student = {

  name: "Shivam Ardeshna",

  age: 18,

  gender: "Male",

  id: "21CE005",

};

function displayStudentDetails(StudentObj) {

  console.log("Name:", StudentObj.name);

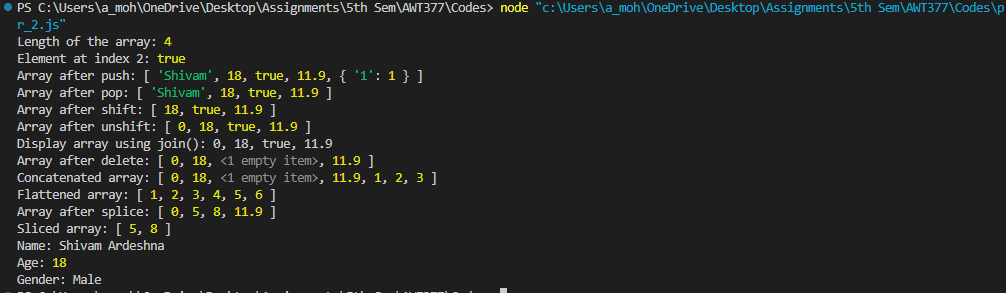
  console.log("Age:", StudentObj.age);

  console.log("Gender:", StudentObj.gender);

}

displayStudentDetails(Student);

**Output:-**

****

**Practical-3**

**Aim:-**

Implement following features of ECMASCRIPT 6.

* The let keyword
* The const keyword
* Arrow Functions
* The (Spread Of) ... Operator
* For/of
* Map Objects
* Set Objects
* Classes
* Promises
* Symbol
* Default Parameters
* Function Rest Parameter

**Code:-**

let x = 6;

console.log(x);

const Name = "Shivam";

console.log(Name);

const add = (a, b) => {

  return a + b;

};

console.log(add(2, 3));

const arr1 = [1, 2, 3];

const arr2 = [...arr1, 4, 5];

console.log(arr2);

const numarr = [1, 2, 3, 4, 5];

for (const num of numarr) {

  console.log(num);

}

const mapObj = new Map();

mapObj.set("Name", "Shivam");

mapObj.set("id", "21CE005");

console.log(mapObj.get("Name"));

console.log(mapObj.has("id"));

mapObj.delete("id");

console.log(mapObj);

const setObj = new Set();

setObj.add(1);

setObj.add(2);

setObj.add(3);

console.log(setObj);

setObj.delete(2);

console.log(setObj);

class Student {

  constructor(name, age, id) {

    this.name = name;

    this.age = age;

    this.id = id;

  }

  Intro() {

    console.log(

      `Bonjour, my name is ${this.name} , I am ${this.age} years old and my CollegeId is ${this.id}.`

    );

  }

}

const std = new Student("SHIVAM", 18, "21CE005");

std.Intro();

const fetchData = () => {

  return new Promise((resolve, reject) => {

    setTimeout(() => {

      const data = "This data from an API";

      resolve(data);

    }, 2000);

  });

};

fetchData()

  .then((data) => console.log(data))

  .catch((error) => console.error(error));

const mySymbol = Symbol("mySymbol");

const obj = {

  [mySymbol]: "This is a Symbol property",

};

console.log(obj[mySymbol]);

function greet(name = "name") {

  console.log(`Hello, ${name}!`);

}

greet();

greet("Shivam Ardeshna");

function multiply(...numbers) {

  let total = 1;

  for (const num of numbers) {

    total \*= num;

  }

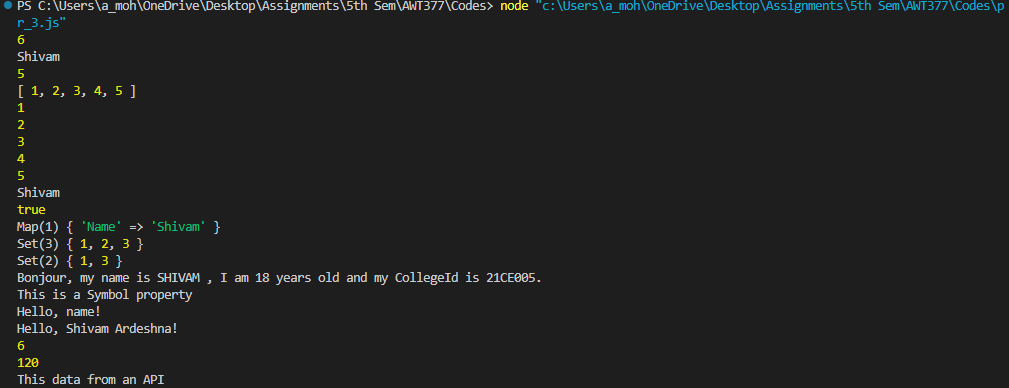
  return total;

}

console.log(multiply(1, 2, 3));

console.log(multiply(1, 2, 3, 4, 5));

**Output:-**

****

**Practical-4**

**Aim:** **Write a function that calculates the factorial of a given number using recursion.**

**Create a nested function that performs a specific task and invoke it within**

**another function.**

**(NOTE: Implement the concept of variable scope in functions by declaring**

**variables with different scopes (global, local) and accessing them).**

**CODE:**

function Factorial(num) {

if (num === 0) {

return 1;

}

return num \* Factorial(num - 1);

}

const number = 5;

console.log(`Factorial of ${number} is:`, Factorial(number));

function Intro(name) {

function IntroPrint() {

return `Hello, ${name}! Welcome to Javascript Tuts`;

}

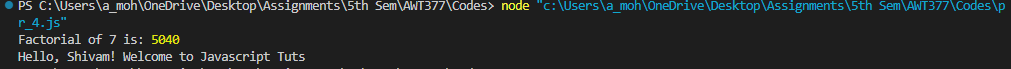
const IntroMessage = IntroPrint();

console.log(IntroMessage);

}

Intro("Shivam");

**Output:**

****