GANPAT UNIVERSITY
U.V. PATEL COLLEGE OF ENGINEERING
B.Tech 5th Semester CE/IT 2CEIT5PE4: Software Packages

Practical-2

Console based node.js applications

1) Create Node.js application which allow users to perform basic mathematical operations such as addition, subtraction, multiplication, and division.

Description:

- Create a Node.js script that takes command-line arguments to perform the desired mathematical operation.
- Implement a logic for each mathematical operation (addition, subtraction, multiplication and division).
- Handle invalid inputs and display appropriate error messages.
- Ensure that the application can handle both integer and floating-point numbers.

CODE:

```
var a=process.argv
var b1=a[2]
var b=parseInt(a[2]);
var c1=a[3];
var c=parseInt(a[3]);
var d=a[4];

switch (d) {
   case '+':
      if (typeof(b)=="number" && typeof(c)=="number") {
         console.log("Addition: "+(b+c));
      }
      else {
         console.log("Enter valid Number.");
      }
      break;
```

```
    case '-':
        if (typeof(b)=="number" && typeof(c)=="number") {
            console.log("subtraction: "+(b-c));
        }
        else {
            console.error("Enter valid Number.");
        }
        break;

    case '*':
        if (typeof(b)=="number" && typeof(c)=="number") {
            console.log("Multiplication: "+(b*c));
        }
        else {
            console.error("Enter valid Number.");
        }
        break;

    case '/':
        if (typeof(b)=="number" && typeof(c)=="number") {
            console.log("Division: "+(b/c));
        }
        else {
            console.error("Enter valid Number.");
        }
        break;

    default:
        console.log("Enter valid Input.")
}
```

OUTPUT:

```
PS E:\SEMwise Materials\SEM-5\SP\Practicals\Prac-1> node 1.js 30 50 -
subtraction: -20
PS E:\SEMwise Materials\SEM-5\SP\Practicals\Prac-1>
```

2) Write a node.js program to build a console application which allow users to perform the following operations:

- Take contact details from user using ReadLine package.
- Add a new contact details like name & phone number (Length should be 10) and store details in an array. Display value of an array.

CODE:

```
const readline = require('readline')
const a1 = readline.createInterface(process.stdin, process.stdout)
a1.question('Enter your Name : ', (name) => {
  a1.question('Enter mobile number : ', (num) => {
    if (num.length == 10) {
      console.log(`Your Name is : ${name}`)
      console.log(`Your Mobile Number is : ${num}`)
    } else {
      console.log(`Enter a valid mobile number.`);
    }
  })
})
```

OUTPUT:

```
Enter your Name : guru
Enter mobile number : 8200297639
Your Name is : guru
Your Mobile Number is : 8200297639
```

3) Write a Node.js program to create an object named book using object literal syntax. Add book_title, author and publish_year as properties to the book object and assign it's appropriate values. Now create function print_info() to print the book object to the console so the final output looks as below:

title: Harry Potter and the Sorcerer's Stone
author: J.K. Rowling publish_year:
1997

CODE:

```
const book = {
   book_title: "Harry Potter and
the Sorcerer's Stone",
   author: "J.K. Rowling",
   publish_year: 1997
}
function print_info() {
   console.log("title: " +
`${book.book_title}`)
   console.log("author: " +
`${book.author}`)
   console.log("publish_year: " +
`${book.publish_year}`)
}
print_info()
```

OUTPUT:

```
PS E:\SEMwise Materials\SEM-5\SP\Practicals\Prac-1> node p2_3.js
title: Harry Potter and the Sorcerer's Stone
author: J.K. Rowling
publish_year: 1997
PS E:\SEMwise Materials\SEM-5\SP\Practicals\Prac-1>
```

4. Create an array named products. Add objects to the array. Each object should be a single product, with 3 properties: name, inventory and unit_price. Create two functions named listProducts() andtotalValue(). A listProducts() function accepts a parameter -- the array of products and it should return an array of the names of the products. A function named totalValue() should accept a parameter -- the array of products and it should return the total value of all of the products in the array. To calculate the total value of one product multiply the inventory value with the unit_price.

CODE:

```
const products = [{
   name: "Pen",
   inventory: 10,
   unit_price: 5
},
{
   name: "Pencil",
   inventory: 5, unit_price:
      5
}, {
   name: "Marker",
   inventory: 10,
   unit_price: 5
}
```

```
];
function listProduct(products) {
   products.forEach(element => {
      console.log('Product: ' + element.name);
   });
}
listProduct(products);
function totalValue(products) {
   var total = 0;
   products.forEach(element => {
      total +=
      element.inventory * element.unit_price;
   });
   console.log("Total value: " + total);
}
totalValue(products);
```

OUTPUT:

```
PS E:\SEMwise Materials\SEM-5\SP
unnerFile.js"
Product: Pen
Product: Pencil
Product: Marker
Total value: 125
```