

Module 1: Assignment

Problem Statement:

Create a simple menu-driven command line application using Nodejs in which the user can enter two numbers and perform addition, multiplication, subtraction, and division.

The requirements look like this:

1. The menu should contain these operations:
 - Addition
 - Subtraction
 - Multiplication
 - Division
 - Exit
2. The application should handle DivideByZero error by showing an error message.
3. The application should exit only when a user chooses the exit option in the menu.

Below is the answer:

```
const readline = require('readline');

// Create an interface for reading user input
const rl = readline.createInterface({
  input: process.stdin,
  output: process.stdout
});

// Function to perform addition
function add(a, b) {
  return a + b;
}

// Function to perform subtraction
function subtract(a, b) {
  return a - b;
}

// Function to perform multiplication
function multiply(a, b) {
  return a * b;
}
```

```
// Function to perform division
function divide(a, b) {
  if (b === 0) {
    throw new Error('DivideByZeroError: Division by zero is not allowed.');
```

```
  }
  return a / b;
}

// Function to display the menu
function displayMenu() {
  console.log('Menu:');
  console.log('1. Addition');
  console.log('2. Subtraction');
  console.log('3. Multiplication');
  console.log('4. Division');
  console.log('5. Exit');
```

```
  }

// Function to handle user input and perform operations
function handleUserInput(choice) {
  switch (choice) {

    case '1':
      rl.question('Enter the first number: ', (num1) => {
        rl.question('Enter the second number: ', (num2) => {
          const result = add(parseFloat(num1), parseFloat(num2));
          console.log('Result:', result);
          displayMenu();
          askForChoice();
        });
      });
      break;

    case '2':
      rl.question('Enter the first number: ', (num1) => {
        rl.question('Enter the second number: ', (num2) => {
          const result = subtract(parseFloat(num1), parseFloat(num2));
          console.log('Result:', result);
          displayMenu();
          askForChoice();
        });
      });
      break;
```

```
case '3':
  rl.question('Enter the first number: ', (num1) => {
    rl.question('Enter the second number: ', (num2) => {
      const result = multiply(parseFloat(num1), parseFloat(num2));
      console.log('Result:', result);
      displayMenu();
      askForChoice();
    });
  });
  break;

case '4':
  rl.question('Enter the first number: ', (num1) => {
    rl.question('Enter the second number: ', (num2) => {
      try {
        const result = divide(parseFloat(num1), parseFloat(num2));
        console.log('Result:', result);
      } catch (error) {
        console.log('Error:', error.message);
      }
      displayMenu();
      askForChoice();
    });
  });
  break;

case '5':
  rl.close();
  break;

default:
  console.log('Invalid choice. Please try again.');
  displayMenu();
  askForChoice();
  break;
}
}
```

```
// Function to ask for user's choice
function askForChoice() {
  rl.question('Enter your choice: ', (choice) => {
    handleUserInput(choice);
  });
}

// Start the application
displayMenu();
askForChoice();
```

Explanation of the code:

The code `readline.createInterface({ input: process.stdin, output: process.stdout })` creates an interface for reading user input from the command line and writing output to the command line.

process.stdin: refers to the standard input stream, which is used to read input from the user.

process.stdout: refers to the standard output stream, which is used to write output to the console.

By creating an interface with these input and output streams, you can use the `rl` object to interact with the user in the command line. It provides methods to read user input and write output to the console.

In the example I provided, we use this `rl` object to read user input when asking for numbers and choices in the command line prompts. It allows us to display the menu, ask for user input, and perform operations based on the user's choices.

Methods explanation:

displayMenu(): display the menu to choose a number for given menu

askForChoice(): takes a number to pass in **handleUserInput(choice)** method

handleUserInput(choice): In that method we had implemented a switch that will decide which method have to call.

In summary:

The implemented Node.js command line application allows users to perform basic arithmetic operations on two numbers. It provides a menu-driven interface with options for addition, subtraction, multiplication, and division. The application handles potential DivideByZero errors and gracefully displays error messages. The program continues running until the user selects the 'Exit' option, ensuring a smooth user experience. Overall, this project showcases key concepts of user input handling, error handling, and menu-driven functionality in Node.js.