Node assessment 1  
INSTRUCTIONS

1. Create a file sum.js in your machine using the following code:
2. /\* ‘Simple mathematics’ program in node.js \*/
3. *console.log(“Assignment1: Sum of two numbers in Node.js”)*
4. Pass two arguments when running the node command

* For example: *$ node sum.js 20 10*

1. Modify the node.js file to perform the operation detailed below:

* Fetch the arguments passed in the CLI
* Store them in variables
* Add the variables
* Print the result

1. Execute the sum.js file using Node.js interpreter.
2. *$ node sum.js*
3. Note: The result should be displayed in the terminal.
4. Exit Node.js

\

assessment 2  
  
INSTRUCTIONS

1. Create a node project folder with a student project name.
2. Verify that the package.json file exists.
3. Use module.exports to export an array of students.
4. For example:
5. [

   {  
           "firstName": "John",  
           "lastName": "Doe"  
          },  
          {  
           "firstName": "Adam",  
           "lastName": "Smith"  
          }  
       ]

1. Assign the parsed content to a variable student
2. Modify the array based on the following instructions:

* Add new element age to John Doe
* key => age
* value => 23
* Iterate over all key-value pairs and test to see if the age exists in each student.
* hasOwnProperty should be true for John, but false for Adam.

1. The result should be displayed in the terminal.
2. Exit Node.js

Assignment 3

INSTRUCTIONS

1. Copy the given code into a file and name it calculate.js  
   const calculate = {

   sum: function(a,b) {

       return a + b;

   },

   subtract: function(a,b) {

       return a - b;

   },

   multiply: function(a,b) {

       return a \* b

   }

}

  module.exports = calculate;

1. Create a test file named calculate.test.js
2. Import the calculate file in the test file: const mathOperations = require('./calculator')
3. Write a test for adding 2 numbers and validate the expected results
4. Write a failing test and see what output you get
5. Write more test cases: for subtract and multiply
6. Run the test in your terminal
7. Observe the test results and coverage from your test cases

