DeCastro, Christine Faith B.
ITBA-4106

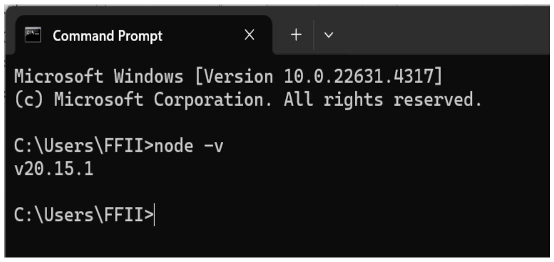
IT 414 – SYSTEMS QUALITY ASSURANCE
LABORATORY ACTIVITY
Automated Unit Testing with Mocha and Chai

Requirements:
1. **Computer**
2. **Internet Connection**
3. **VS Code**

Instructions:

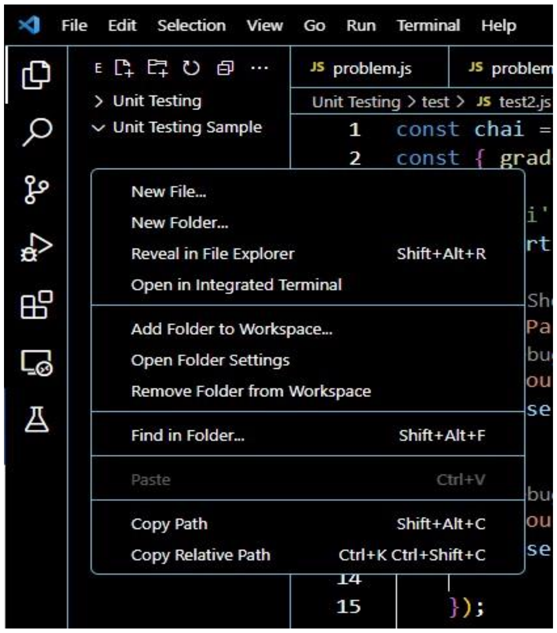
1. Check if node.js is already installed on your machine by opening the command prompt and typing 'node -v'. If a version number shows up, it means it is already installed. If not, go to 'https://nodejs.org/en' and download and install.

   Answer:



```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\FFII>node -v
v20.15.1

C:\Users\FFII>
```

2. Create a folder with your last name as the folder name.
3. Open VS Code and add that folder to the workspace by right clicking on the Explorer tab.

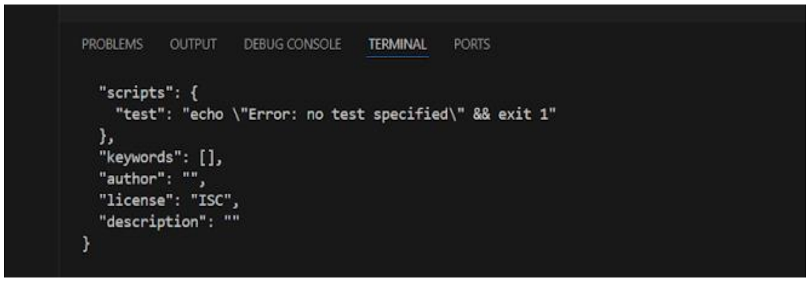

File
Edit
Selection
View
Go
Run
Terminal
Help
problem.js
Unit Testing
Unit Testing Sample
Unit Testing > test > test2.js
New File...
New Folder...
Reveal in File Explorer Shift+Alt+R
Open in Integrated Terminal
Add Folder to Workspace...
Open Folder Settings
Remove Folder from Workspace
Find in Folder... Shift+Alt+F
Paste Ctrl+V
Copy Path Shift+Alt+C
Copy Relative Path Ctrl+K Ctrl+Shift+C

4. Right click on the name of your folder and select Open in Integrated Terminal.

DeCastro, Christine Faith B.
ITBA-4106

5. On the Terminal, type in 'npm init -y' and hit enter to initialized a node.js project.

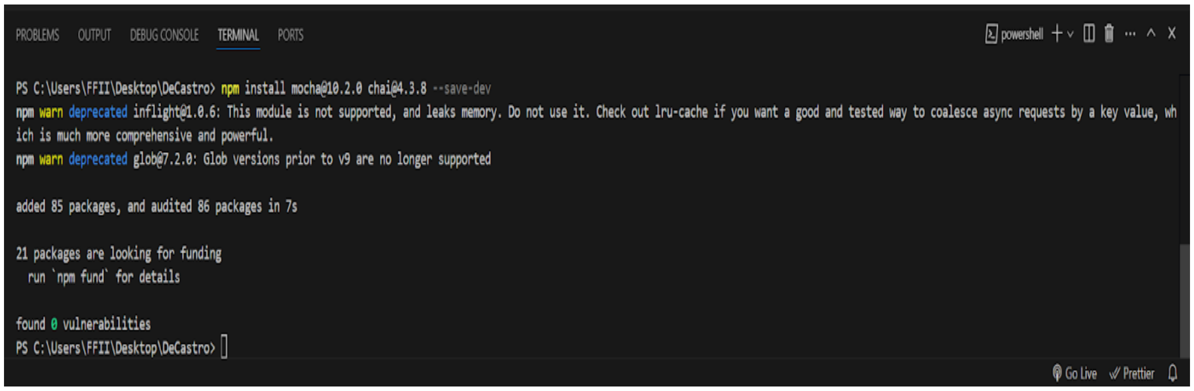
Answer:



```
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": ""
}
```

6. Then type in 'npm install mocha@10.2.0 chai@4.3.8 --save-dev' then wait for the dependencies to be installed.

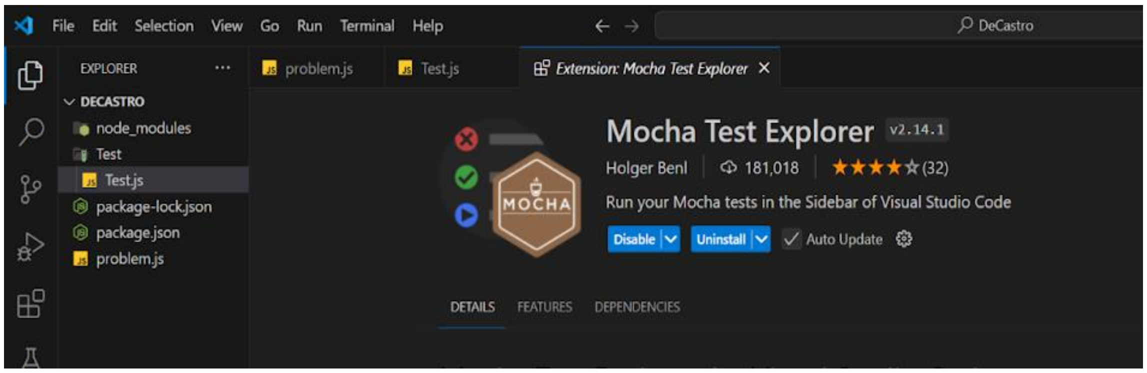
   Answer:



```
PS C:\Users\FFII\Desktop\DeCastro> npm install mocha@10.2.0 chai@4.3.8 --save-dev
npm warn deprecated inflight@1.0.6: This module is not supported, and leaks memory. Do not use it. Check out lru-cache if you want a good and tested way to coalesce async requests by a key value, wh
ich is much more comprehensive and powerful.
npm warn deprecated glob@7.2.0: Glob versions prior to v9 are no longer supported

added 85 packages, and audited 86 packages in 7s

21 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\Users\FFII\Desktop\DeCastro>
```

7. In your project folder create a file named 'problem.js'
8. Then create another folder inside your project folder named 'test' then inside that folder create a new file named 'test.js'
9. Click on "Extensions" on the left-hand side of the screen and search and install the extension named 'Mocha Test Explorer'.

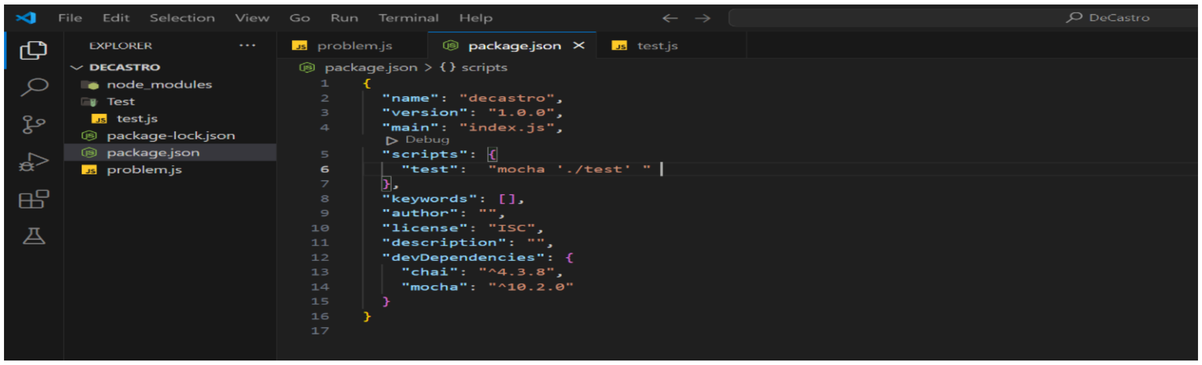
Answer:



Mocha Test Explorer v2.14.1
Holger Benl | 181,018 | ★★★★☆ (32)
Run your Mocha tests in the Sidebar of Visual Studio Code

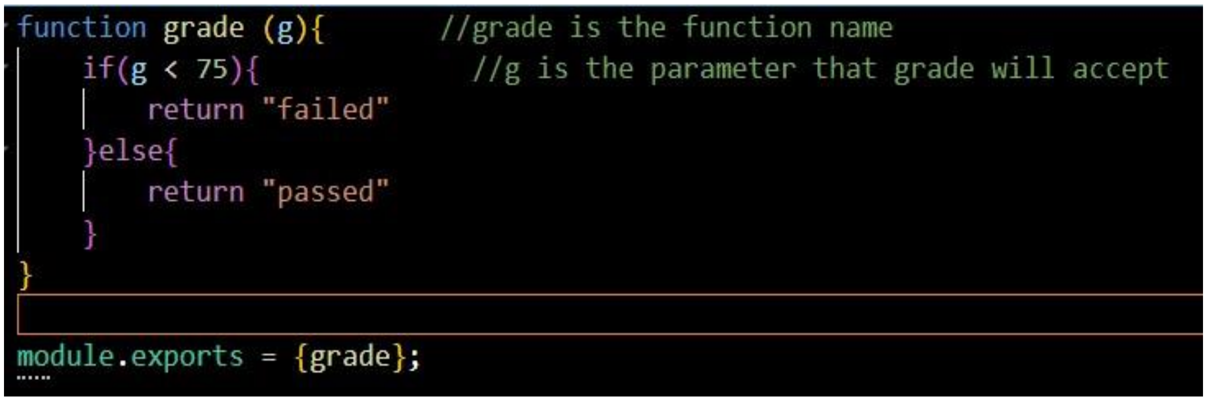
DeCastro, Christine Faith B.
ITBA-4106

10. in the package.json file, replace the value of test with "test": *"mocha './test' "*
    answer:



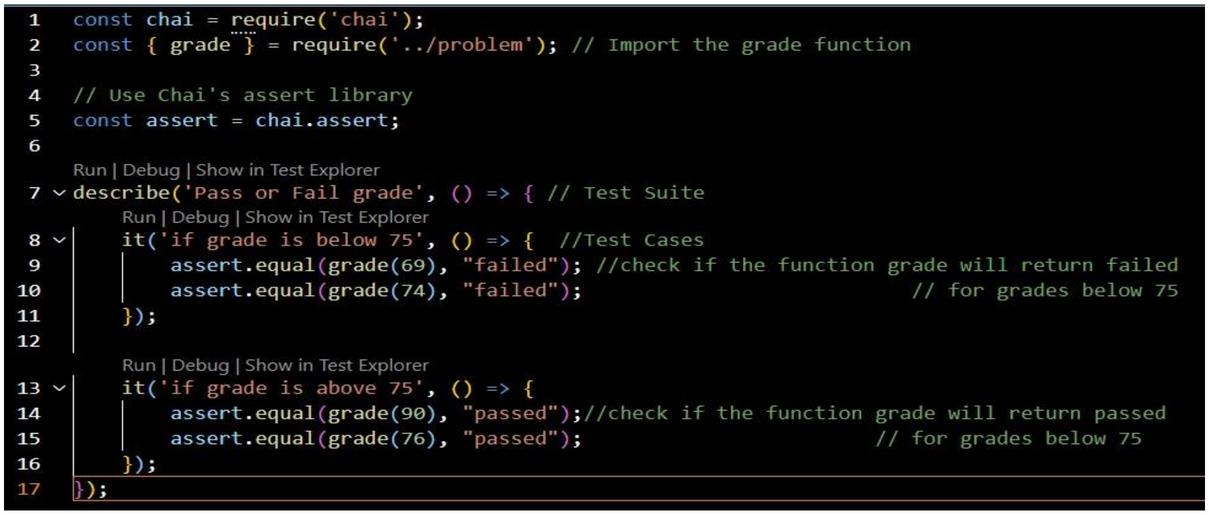
```
{
  "name": "decastro",
  "version": "1.0.0",
  "main": "index.js",
  "scripts": {
    "test":  "mocha './test' "
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "description": "",
  "devDependencies": {
    "chai": "^4.3.8",
    "mocha": "^10.2.0"
  }
}
```

11. Type in the following function in the ‘problem.js’ file:



```
function grade (g){        //grade is the function name
    if(g < 75){             //g is the parameter that grade will accept
        return "failed"
    }else{
        return "passed"
    }
}

module.exports = {grade};
```

12. Type in the following in the ‘test.js’ file:


Run | Debug | Show in Test Explorer
Run | Debug | Show in Test Explorer
Run | Debug | Show in Test Explorer

```
const chai = require('chai');
const { grade } = require('../problem'); // Import the grade function

// Use Chai's assert library
const assert = chai.assert;

describe('Pass or Fail grade', () => { // Test Suite
    it('if grade is below 75', () => {  //Test Cases
        assert.equal(grade(69), "failed"); //check if the function grade will return failed
        assert.equal(grade(74), "failed");                                  // for grades below 75
    });

    it('if grade is above 75', () => {
        assert.equal(grade(90), "passed");//check if the function grade will return passed
        assert.equal(grade(76), "passed");                               // for grades below 75
    });
});
```

13. Run the tests using the ‘Testing’ tab on the left hand side of the screen.
14. Modify the function so that it will have the following return value for each value range of grade.

Return value	Value range

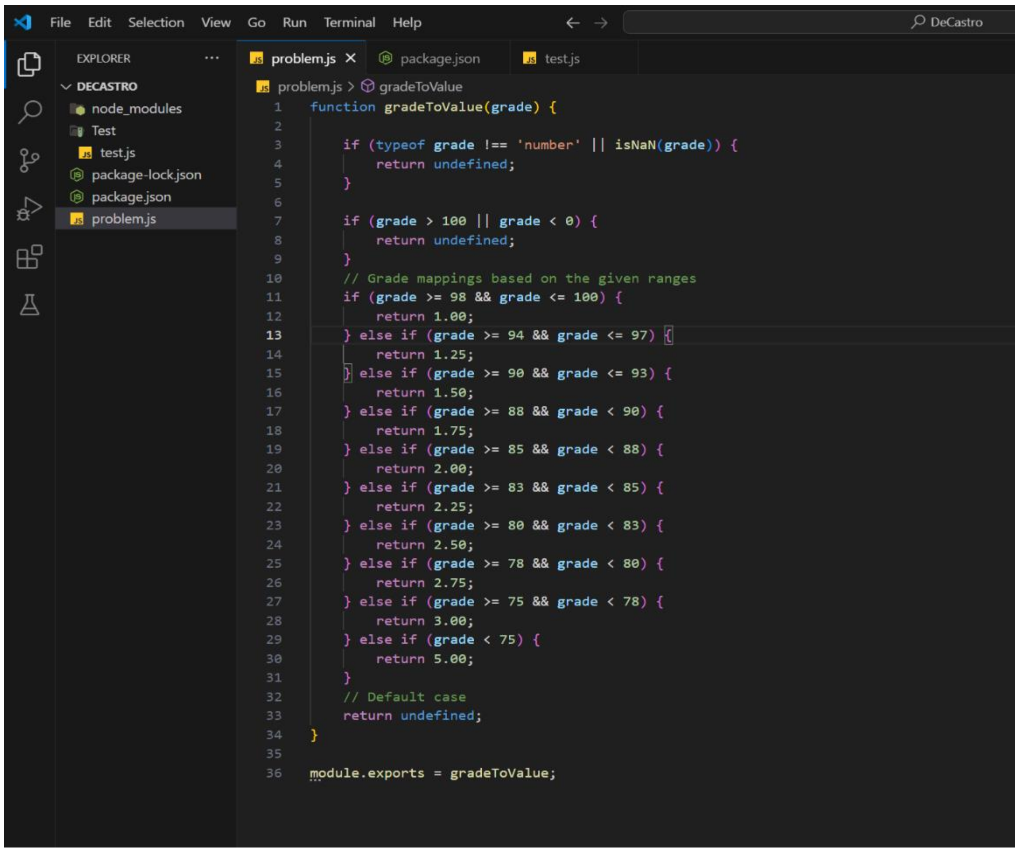
DeCastro, Christine Faith B.
ITBA-4106

1.00	98	-	100
1.25	94	-	97
1.5	90	-	93
1.75	88	-	89
2.00	85	-	87
2.25	83	-	84
2.50	80	-	82
2.75	78	-	79
3.00	75	-	77
5.00	Below 70		

15. Then make at least three test cases in "test.js" file for each return value. (total Test Cases should be 30).

Answer:

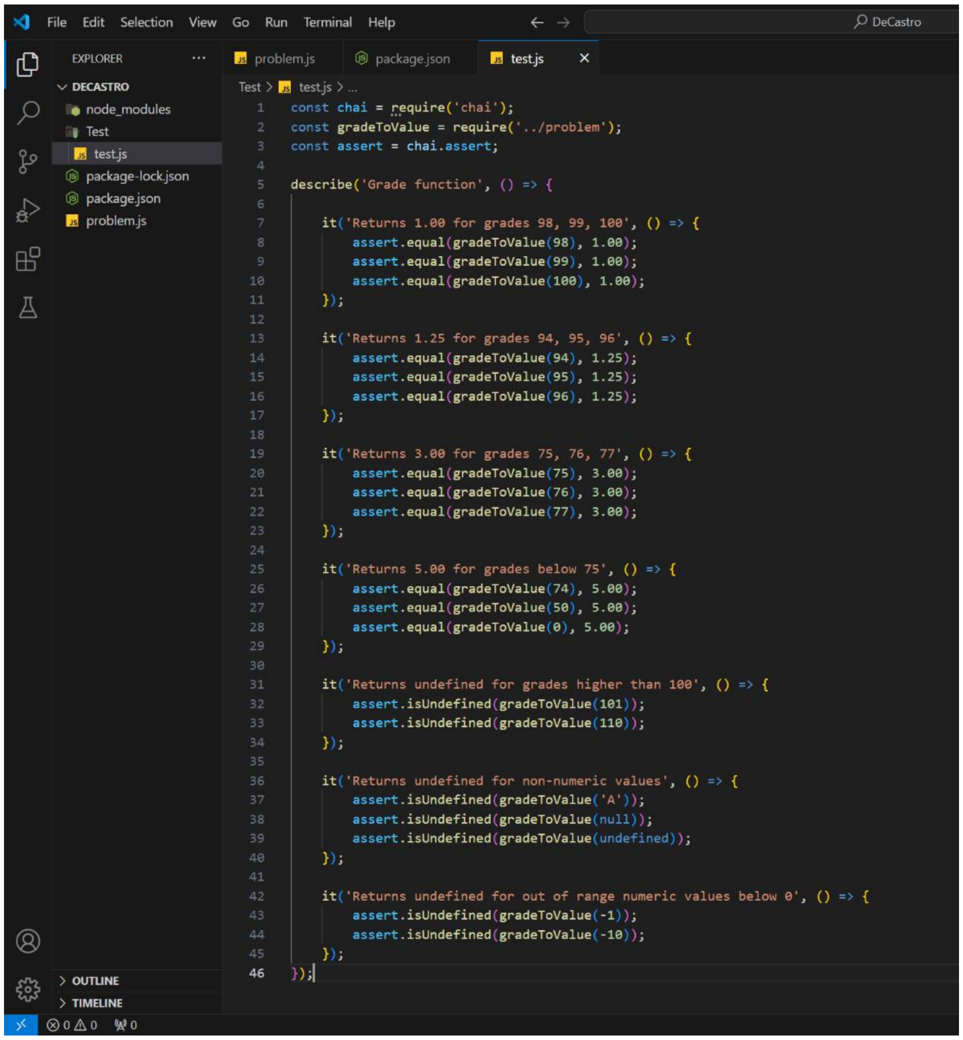
"problem.js"



```
function gradeToValue(grade) {

    if (typeof grade !== 'number' || isNaN(grade)) {
        return undefined;
    }

    if (grade > 100 || grade < 0) {
        return undefined;
    }
    // Grade mappings based on the given ranges
    if (grade >= 98 && grade <= 100) {
        return 1.00;
    } else if (grade >= 94 && grade <= 97) {
        return 1.25;
    } else if (grade >= 90 && grade <= 93) {
        return 1.50;
    } else if (grade >= 88 && grade < 90) {
        return 1.75;
    } else if (grade >= 85 && grade < 88) {
        return 2.00;
    } else if (grade >= 83 && grade < 85) {
        return 2.25;
    } else if (grade >= 80 && grade < 83) {
        return 2.50;
    } else if (grade >= 78 && grade < 80) {
        return 2.75;
    } else if (grade >= 75 && grade < 78) {
        return 3.00;
    } else if (grade < 75) {
        return 5.00;
    }
    // Default case
    return undefined;
}

module.exports = gradeToValue;
```

"Test.js"

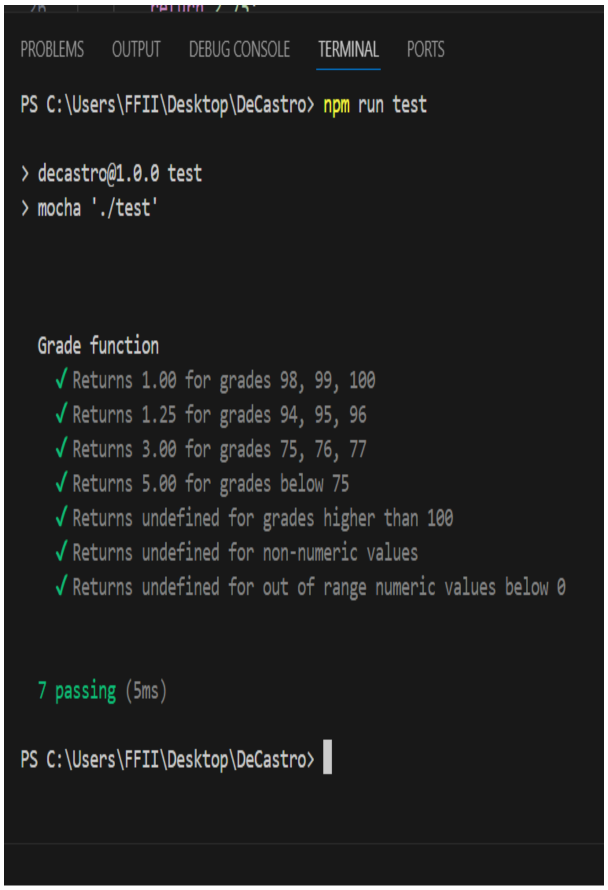
DeCastro, Christine Faith B.
ITBA-4106



```javascript
const chai = require('chai');
const gradeToValue = require('../problem');
const assert = chai.assert;

describe('Grade function', () => {

    it('Returns 1.00 for grades 98, 99, 100', () => {
        assert.equal(gradeToValue(98), 1.00);
        assert.equal(gradeToValue(99), 1.00);
        assert.equal(gradeToValue(100), 1.00);
    });

    it('Returns 1.25 for grades 94, 95, 96', () => {
        assert.equal(gradeToValue(94), 1.25);
        assert.equal(gradeToValue(95), 1.25);
        assert.equal(gradeToValue(96), 1.25);
    });

    it('Returns 3.00 for grades 75, 76, 77', () => {
        assert.equal(gradeToValue(75), 3.00);
        assert.equal(gradeToValue(76), 3.00);
        assert.equal(gradeToValue(77), 3.00);
    });

    it('Returns 5.00 for grades below 75', () => {
        assert.equal(gradeToValue(74), 5.00);
        assert.equal(gradeToValue(50), 5.00);
        assert.equal(gradeToValue(0), 5.00);
    });

    it('Returns undefined for grades higher than 100', () => {
        assert.isUndefined(gradeToValue(101));
        assert.isUndefined(gradeToValue(110));
    });

    it('Returns undefined for non-numeric values', () => {
        assert.isUndefined(gradeToValue('A'));
        assert.isUndefined(gradeToValue(null));
        assert.isUndefined(gradeToValue(undefined));
    });

    it('Returns undefined for out of range numeric values below 0', () => {
        assert.isUndefined(gradeToValue(-1));
        assert.isUndefined(gradeToValue(-10));
    });
});
```

“npm run test”

DeCastro, Christine Faith B.
ITBA-4106



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\FFII\Desktop\DeCastro> npm run test

> decastro@1.0.0 test
> mocha './test'



  Grade function
    ✓ Returns 1.00 for grades 98, 99, 100
    ✓ Returns 1.25 for grades 94, 95, 96
    ✓ Returns 3.00 for grades 75, 76, 77
    ✓ Returns 5.00 for grades below 75
    ✓ Returns undefined for grades higher than 100
    ✓ Returns undefined for non-numeric values
    ✓ Returns undefined for out of range numeric values below 0


  7 passing (5ms)

PS C:\Users\FFII\Desktop\DeCastro>
```

You will be graded based on the following rubrics.

DeCastro, Christine Faith B.
ITBA-4106

PROGRAM (20)	EXCELLENT	GOOD	FAIR	POOR
Program Execution	Program executes correctly with no syntax or runtime errors(5)		Program executes with a minor (easily fixed error) (2)	Program does not execute(1)
Correct Output	Program displays correct output with no errors (5)	Output has minor errors(4)	Output has multiple errors(3)	Output is incorrect(2)
Design of Logic	Program is logically well designed (5)	Program has slight logic errors that do not significantly affect the results(4)	Program has significant logic errors(3)	Program is incorrect(2)
Standards	Program is stylistically well designed(5)	Few inappropriate design choices(i.e. poor variable names, improper indention)(4)	Several inappropriate design choice (i.e. poor variable names, improper indention)s(3)	Program is poorly written(2)

ANSWER: