Help to complete the tasks of this exercise can be found on the chapters from ch. 3 ” Creating a Node.js Module” and ch. 4 “Building a Simple Web Server” of our course book “Get Programming with Node.js” by Jonathan Wexler. The aims of the exercise are to teach to understand and use Node.js modules and give to the skills to build a very simple web server.

Embed your theory answers, drawings, codes and screenshots directly into this document. Always immediately after the relevant question. Return the document in itslearning by the deadline.

It’s also recommendable to use Internet sources to supplement the information provided by the course book.

The maximum number of points you can earn from this exercise is 10.

Tasks:

1. Explain benefits of using modules in Node.js (1 point)
2. Visit the web site https://docs.npmjs.com/. Answer the questions below. (2 points)
3. What are the three main distinct components of npm? (0.5 points)
4. For what can npm be used for? (0.5 points)
5. Write npm CLI command to
   1. create a package.json file for your application
   2. install a package to your application so that also the dependencies in package.json are updated
   3. install a package globally on your computer
   4. run the javascript file that is specified in the start property of a scripts object in a package.json file
   5. remove a package

(1 point)

1. Program a Node.js module “evaluator”, which can be used to give a grade for a student by using a custom evaluation scale. (3 points)

The module exports two functions. The first function’s signature is setEvaluationScale(scale). The argument scale is of a type array.

The example below clarifies the structure of the scale argument.

[{ grade: 1, points: 20},{ grade: 2, points: 35},

{ grade: 3, points: 50},

{ grade: 4, points: 65},{ grade: 5, points: 80}]

The feature points is the minimum number of points that the student needs to collect to get the grade given in the feature grade. In case all the points are more than the student has collected, then the student gets the grade 0.

The second function’s signature is getGrade(points). The argument points is the number of points a student has collected. The functions returns the grade for the given points.

Please note that if someone tries to call function getGrade before the evaluation scale is set, then a message ‘There is no evaluation scale defined.’ is returned.

Require the module and test that it functions as expected by calling its methods.

1. Working with a simple web server. (2 points)

a. Program and run the web server presented below. Test it by using a browser. (0.5 points)

const http = require("http");

const httpStatus = require("http-status-codes"); //1

const port = 3000; //2

const app = http.createServer((request, response) => { //3

console.log("Received an incoming request!");

response.writeHead(httpStatus.OK, { //4

"Content-Type": "text/html"

});

let responseMessage = "<h1>Hello, Universe!</h1>";

response.write(responseMessage); //5

response.end(); //6

console.log(`Sent a response : ${responseMessage}`);

});

app.listen(port); //7

1. Explain carefully what happens in the statements marked with comments 1 to 8 on at a time. (1 point)
2. A student has received 81 points on a course. Use the module from the previous task, the web server above, and print the grade on a web page. (0.5 points)
3. Asynchronous programming. (4 \* 0,5 = 2 points, you can leave one unanswered)
4. Program the blocking code and non-blocking code examples that can be found at the address <https://www.tutorialspoint.com/nodejs/nodejs_callbacks_concept.htm>.
5. Execute the examples and explain the possible differences in results.
6. Explain the concept of a callback function.
7. What part of the code is a callback function?
8. How the concept of non-blocking is relevant here?