**Full Stack Development III – Lab 2**

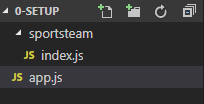
**Node Fundamentals I**

**Developer Note:**When updating your code in javascript files, you need to stop Node from running using Ctrl C twice and then run the Node command again at the command line ie.**> node app.js** for your changes to take effect. Node will not see your current changes, as previous code has been compiled down to machine code and it is running it.

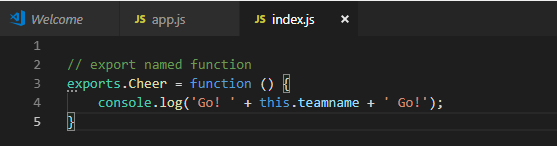
**NodeJs: Module, Export, Require**

**Exercise #1.**

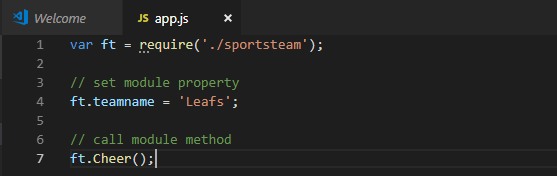
1. Open a command prompt create a directory for exercise 1
2. Open Visual Studio Code and open the folder exercise 1
3. Create a new file named app.js file.   
    Then create a folder named **sportsteam**.   
    Next create an index.js, which is the module entry point.

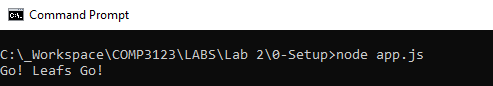


1. In the index.js, write the following code to export a named function as a module.



1. In the app.js file, write the following code to require and use the module.

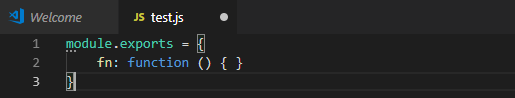


1. In the command prompt, use node to execute the program.  
   
2. Expand this module to declare two methods, **Boo()** and **Cheer()** and call them from **app.js**

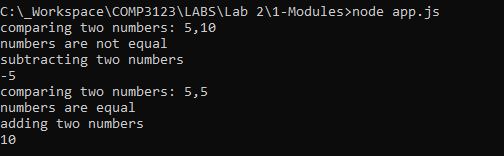
**Exercise #2:**

* 1. Create two modules named **calculator** and **comparer**.
* Comparer will export a function named **AreNumberEqual**, which will compare two numbers and return true or false, if equal or not equal.
* Calculate will export an anonymous object, with two methods **Add()** and **Subtract().**

Remember a module that exports an object has this signature.



* 1. In the main app.js, import both modules using require. Then use them in the file to call the function **AreNumbersEqual** in the Comparer module. The method will take two numbers as parameters with the following requirements:
     + When numbers are equal then add them
     + When numbers are not equal then subtract them.
  2. The following is the expected output in the CLI.



**NodeJs: Using the Events and Emitter**

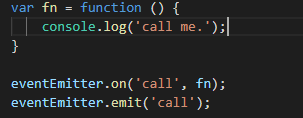
**Exercise #3:**

1. Using require include the event module and use to the **Event Emitter** to trigger using the **on()** and **emit()** events.e

<https://www.npmjs.com/package/EventEmitter>



2. Create an Event Handler to handle when the event is raised. Event handler written as a function expression has the following signature. It can be wired up with events in the following way.



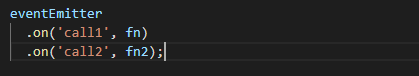
3. Using **require** and the **Event Emitter**, create an **Event Handler** to raise an alarm and trigger it using the **emit()** keyword. The console output is below:



4. Create a second event handler that will be emitted when the first alarm handler is executed.



5. Try using event chaining, instead of two separate calls to using Event Emitter



**Node as a Web Server**

**Exercise #4:**

1. Create a new directory and then create an app.js file. Open your workspace in Visual Studio code. In the app.js file write or copy the following code to create your first Web Server with Node.js.

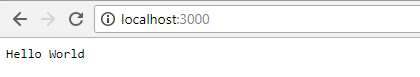
\*\* Startup Code can be copied at the following link below:  
<https://nodejs.org/en/about/>

|  |
| --- |
| **const** http = require('http');  **const** hostname = '127.0.0.1'; **const** port = 3000;  **const** server = http.createServer((req, res) => {  res.statusCode = 200;  res.setHeader('Content-Type', 'text/plain');  res.end('Hello World\n'); });  server.listen(port, hostname, () => {  console.log(`Server running at http://${hostname}:${port}/`); }); |

1. Run the app.js at the command line with node. The server will start up and be waiting to handle requests.



1. Open a web browser and navigate to local host port 3000 to trigger a response to the running web server.



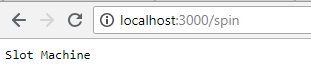
**Exercise #5: Slot Machine**

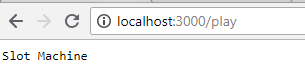
1. Create a web server with the previous code. Handle two different requests using

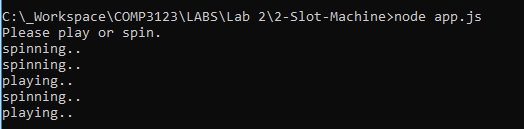
**req.url** and return the matching response when changing the query string path.

An unknown request will return the message ‘Please play or spin’ in the console

output.







**Challenge: Slot Machine – Part 2**

1. Add the reference to built-in **url module** using require  
   
2. Use the documentation to **parse** the **url** and compare the **query string** path.
3. Execute the following request in the browser, the server should respond in the console with the expected output below.

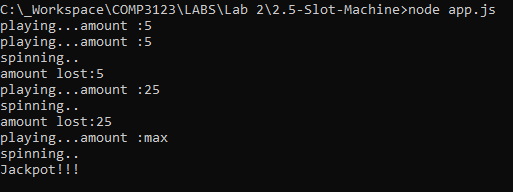
Hint: You will need a variable to track the amount between requests. Play will set amount and spin will output it.







4. Update your code to use the event emitter, so that when the query string **amount=max**, an event is emitted and an event handler will trigger and output the text **‘Jackpot!’** to the console.