Lab tutorial How Module works in JS :

CN5006 Web And Mobile Development

Dr.Nadeem Qazi

---

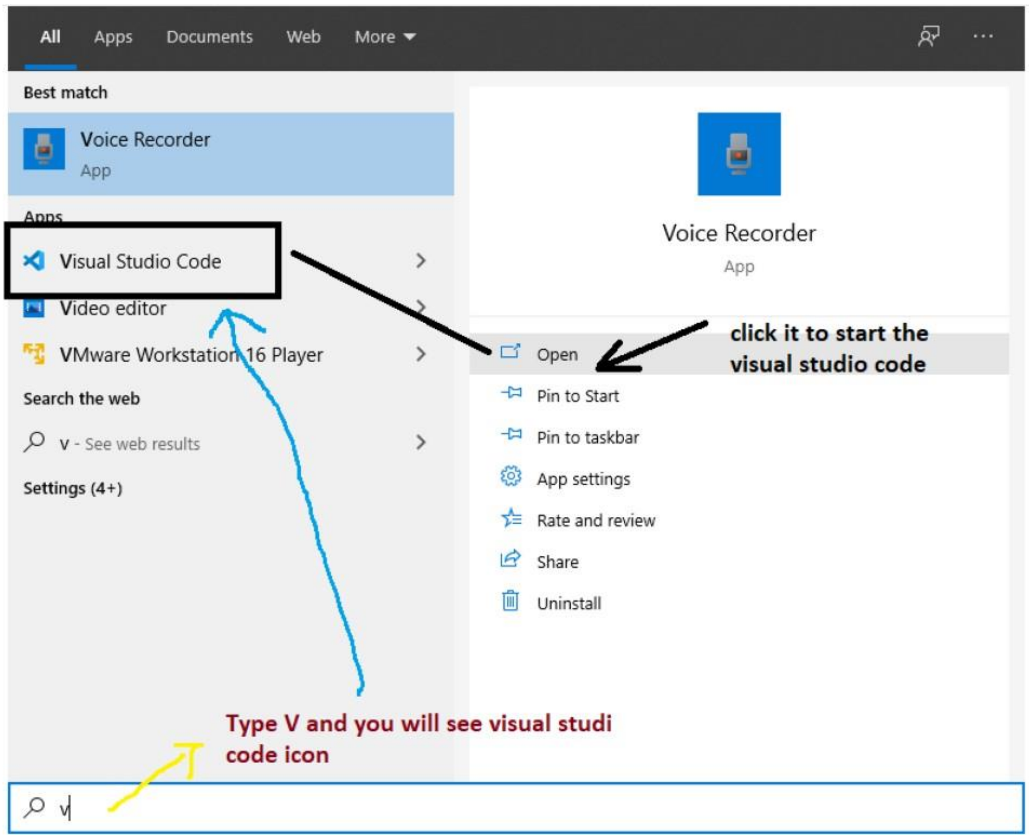
1. **Lab Session**
   1. Functions in JavaScript:
      a. What is JS functions
      b. How to write functions in Java script
      c. How to export a function with argument.
      d. Defining an Arrow function

2. How to Create Local Module in NodeJS
   - How to export a function/s and variable/s to other module
   - How to use a module declared in a separate file
3. Access NodeJS Core Module

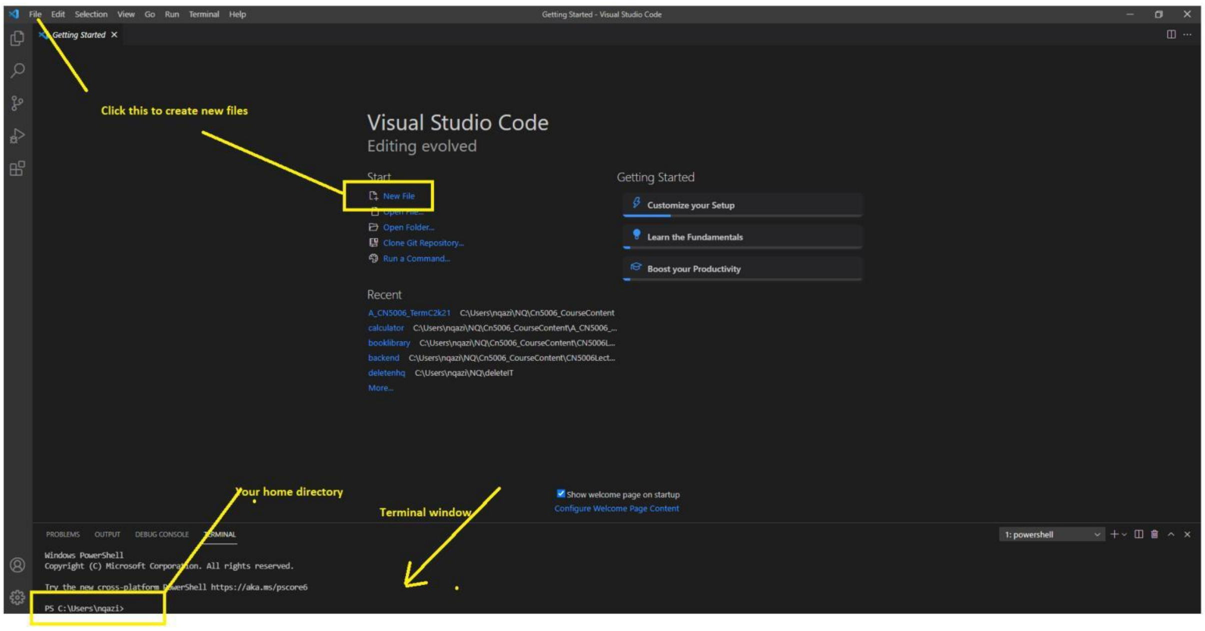
before you write the code please note // is used to comment in JS and it is NOT treated as CODE

Step 1 Make sure you have installed Visual Studio Code and NodeJS.

Step 2: Open Visual Studio code by typing V as shown in the figure below



Once the visual studio code is started you will see the following window :

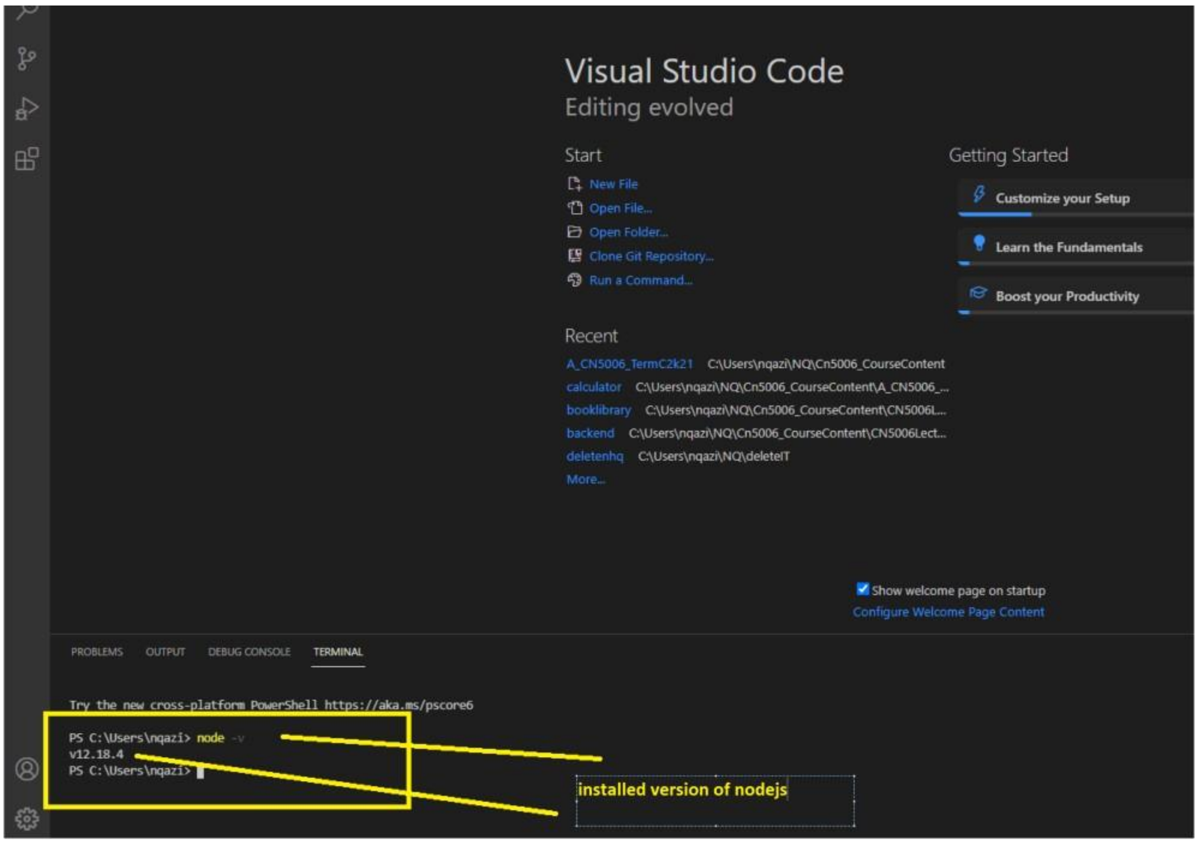


```
A_CN5006_TermC2k21   C:\Users\nqazi\NQ\Cn5006_CourseContent
calculator   C:\Users\nqazi\NQ\Cn5006_CourseContent\A_CN5006_...
booklibrary   C:\Users\nqazi\NQ\Cn5006_CourseContent\CN5006L...
backend   C:\Users\nqazi\NQ\Cn5006_CourseContent\CN5006Lect...
deletenhq   C:\Users\nqazi\NQ\deleteIT
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\nqazi>
```

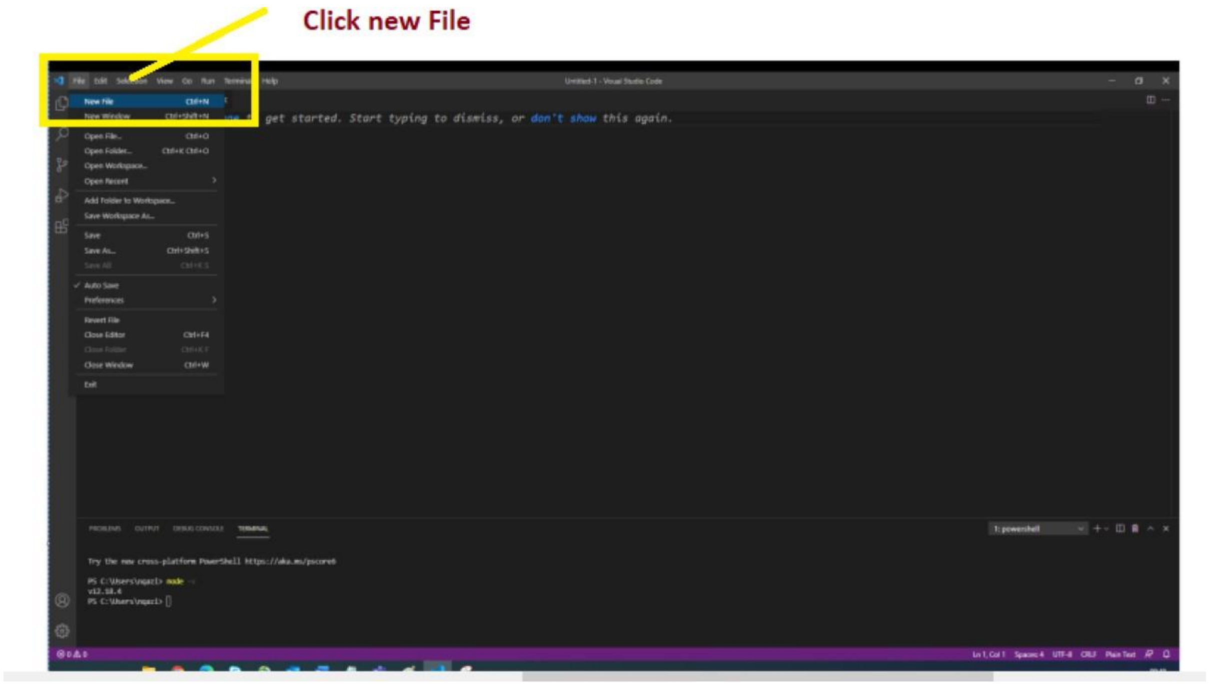
Step 3 check if node JS is installed. In the terminal window , on the prompt write following line :**node -v**

you should see the version number of the nodeJS running on your system as shown in the figure below, however you may have different version of nodeJS running


Visual Studio Code
Editing evolved
Start
New File
Open File...
Open Folder...
Clone Git Repository...
Run a Command...
Getting Started
Customize your Setup
Learn the Fundamentals
Boost your Productivity
Recent
A_CN5006_TermC2k21 C:\Users\nqazi\NQ\Cn5006_CourseContent
calculator C:\Users\nqazi\NQ\Cn5006_CourseContent\A_CN5006_...
booklibrary C:\Users\nqazi\NQ\Cn5006_CourseContent\CN5006L...
backend C:\Users\nqazi\NQ\Cn5006_CourseContent\CN5006Lect...
deletenhq C:\Users\nqazi\NQ\deleteIT
More...
Show welcome page on startup
Configure Welcome Page Content
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
installed version of nodejs

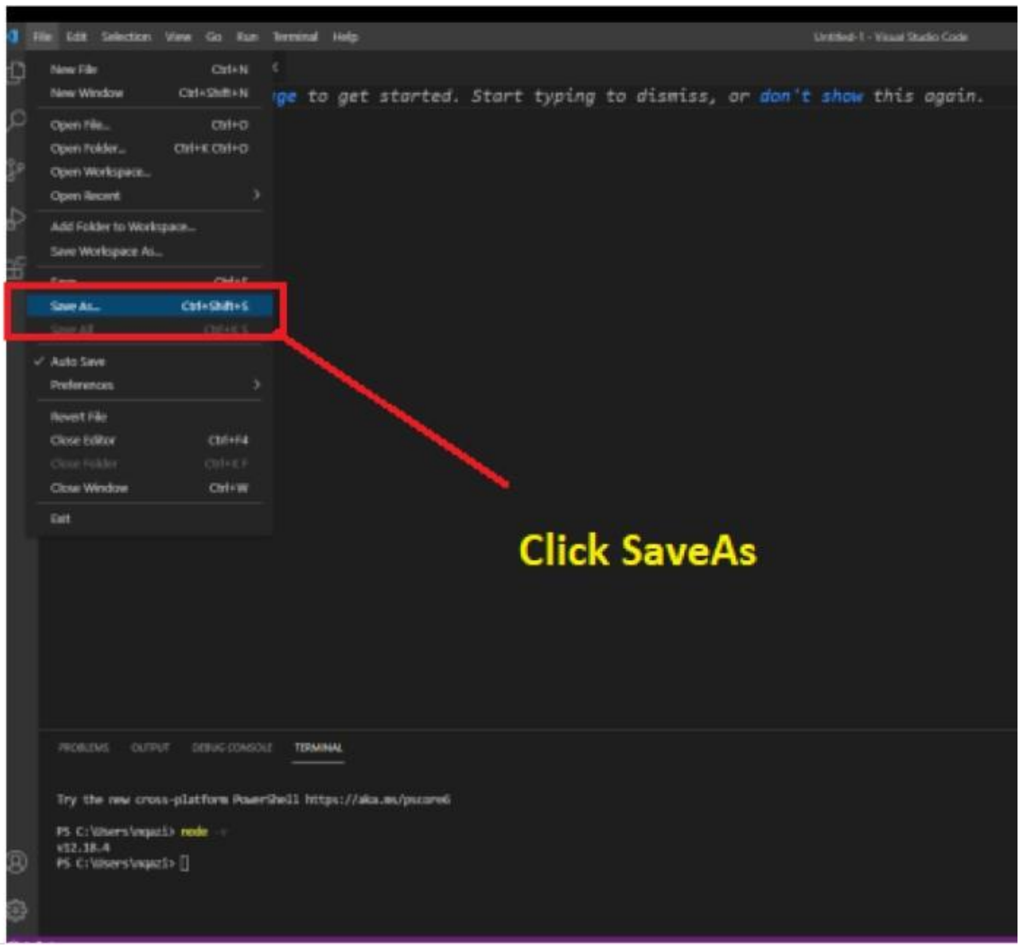
```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\nqazi> node -v
v12.18.4
PS C:\Users\nqazi>
```

Now you are all set to start coding in Visual Studio Code. Click File ,new to create a new file as shown in the figure below



```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\nqazi> node -v
v12.18.4
PS C:\Users\nqazi>
```

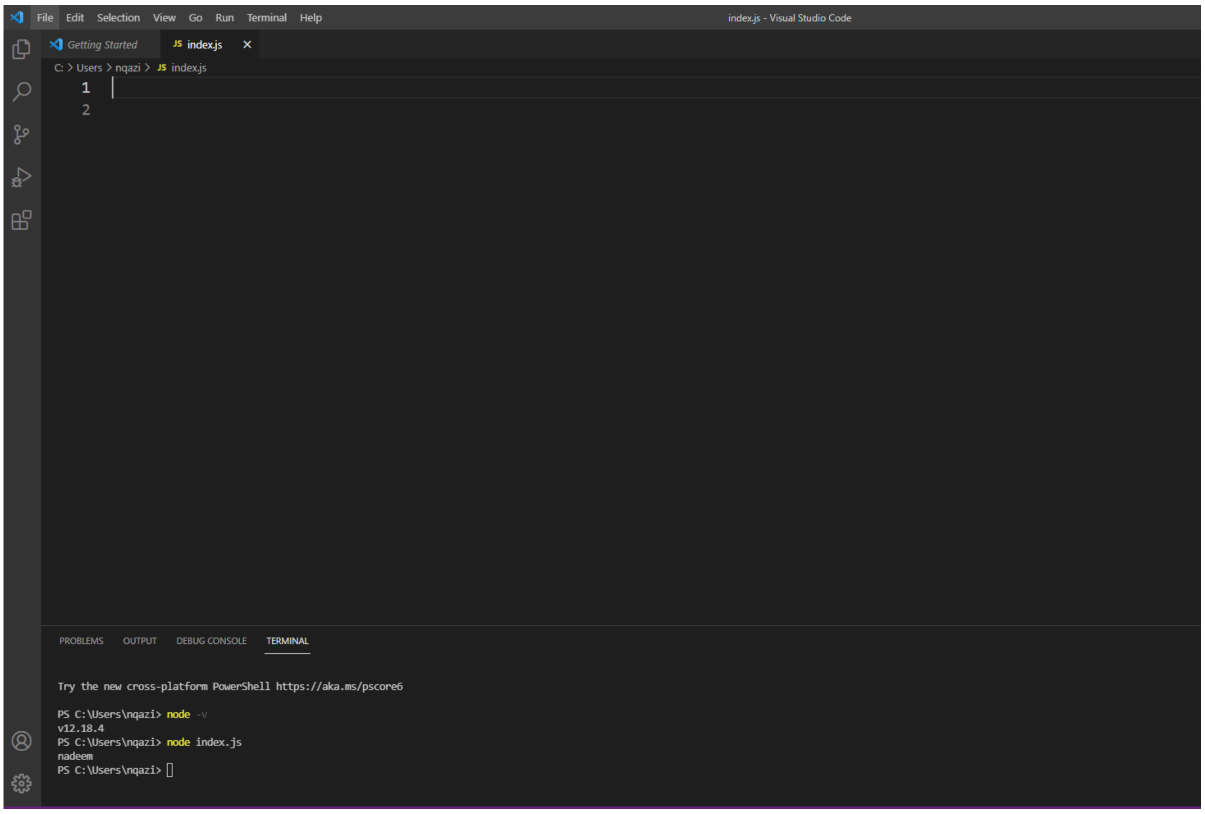
It will open a new blank file , save this file as Java script file by clicking SaveAS



```
ge to get started. Start typing to dismiss, or don't show this again.
```

```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\nqazi1> node -v
v12.18.4
PS C:\Users\nqazi1> 
```

It will open a dialogue box Choose the name of the file as index.js which will then create a index.js file for you to start coding . as shown in the figure below



```
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\nqazi> node -v
v12.18.4
PS C:\Users\nqazi> node index.js
nadeem
PS C:\Users\nqazi> 
```

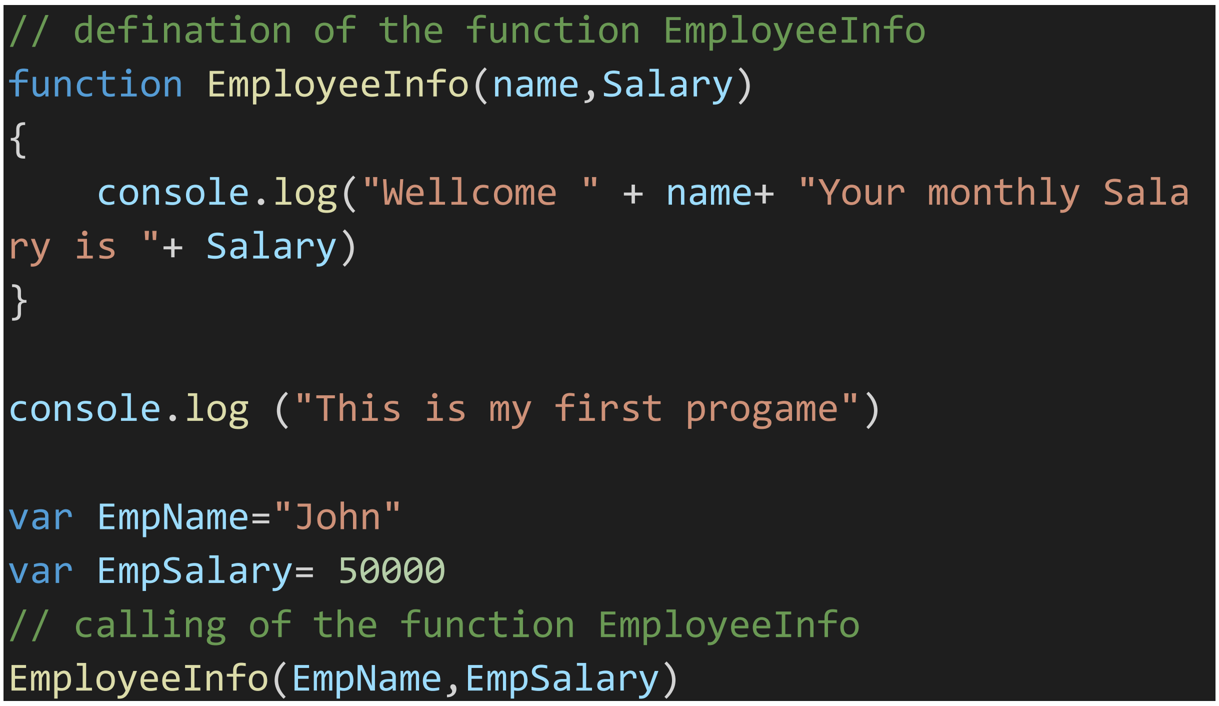
<u>Now Star writing the code</u>

<u>Exercise 1:</u>

<u>Exercise 1 requires you to write the code in a file name it index.js</u>

Follow these steps to write and save a JavaScript file (index.js) that defines a function called <u>**`Employeeinfo.js`**</u> and calls this function with different parameter values.

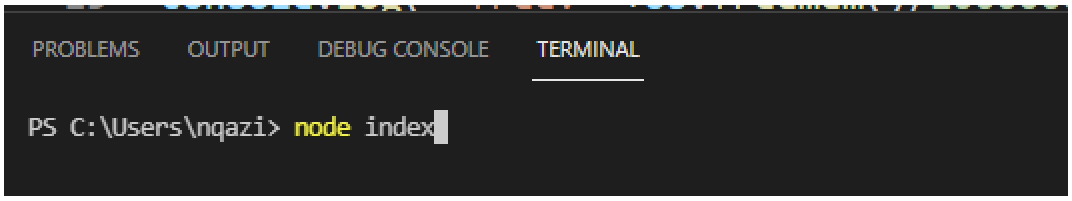
- **Create a Index.js File:** In your editor, create a new file and name it `index.js`.
- **Write the Code:**
  - Inside the `index.js` file, define a function named `Employeeinfo` that accepts two parameters: `name` and `salary`.
  - Call this function with different values for the parameters. The code is shown below:, write the code and save the index.js file



```
// defination of the function EmployeeInfo
function EmployeeInfo(name,Salary)
{
    console.log("Wellcome " + name+ "Your monthly Sala
ry is "+ Salary)
}

console.log ("This is my first progame")

var EmpName="John"
var EmpSalary= 50000
// calling of the function EmployeeInfo
EmployeeInfo(EmpName,EmpSalary)
```

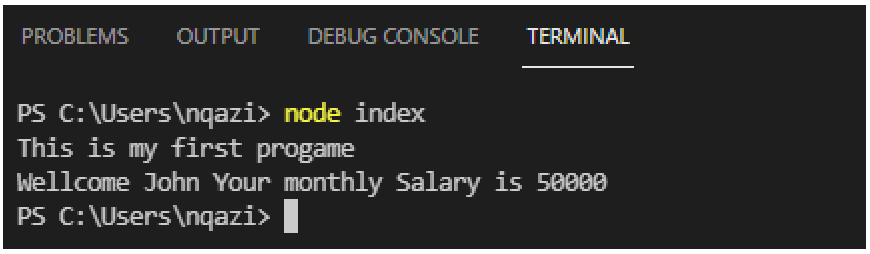
How to run this Code:

In the terminal window type " node index" and hit enter.



```
PS C:\Users\nqazi> node index
```

You will see you code will be executed and the following message will appear in the Terminal window.



```
PS C:\Users\nqazi> node index
This is my first progame
Wellcome John Your monthly Salary is 50000
PS C:\Users\nqazi>
```

Code Explanation

1. **console.log** is used to display a message or variable value in the terminal window, so when you use in your code : console.log("This is my first programme"), this is displayed in the terminal window. Note the use of quotes "" to display the message.
2. You define the function in Java script with the keyword function followed by name of the function which in this case is EmployeeInfo , (you can give any name to the function) .
3. The parameter/s of the function is defined in parentheses () which in this

case, is (name,Salary), so the complete signature of the function is :

```
function EmployeeInfo(name,Salary)
{
    console.log("Wellcome " + name+ "Your monthly Sa
lary is "+ Salary)
}
```

Observe the use of “+” in console.log to concatenate message with the value of parameters passed in this function i.e. name and Salary.

Exercise 2 : Create Arrow Functions:

Arrow functions are a modern way of defining the functions in Javascript, Unlike traditional JavaScript functions, they do not require a name and are more concise. To learn more for watch the Arrow functions video in the lecture uploaded on Moodle.

Syntax of the ArrowFunction:

const variablename= () =>{}

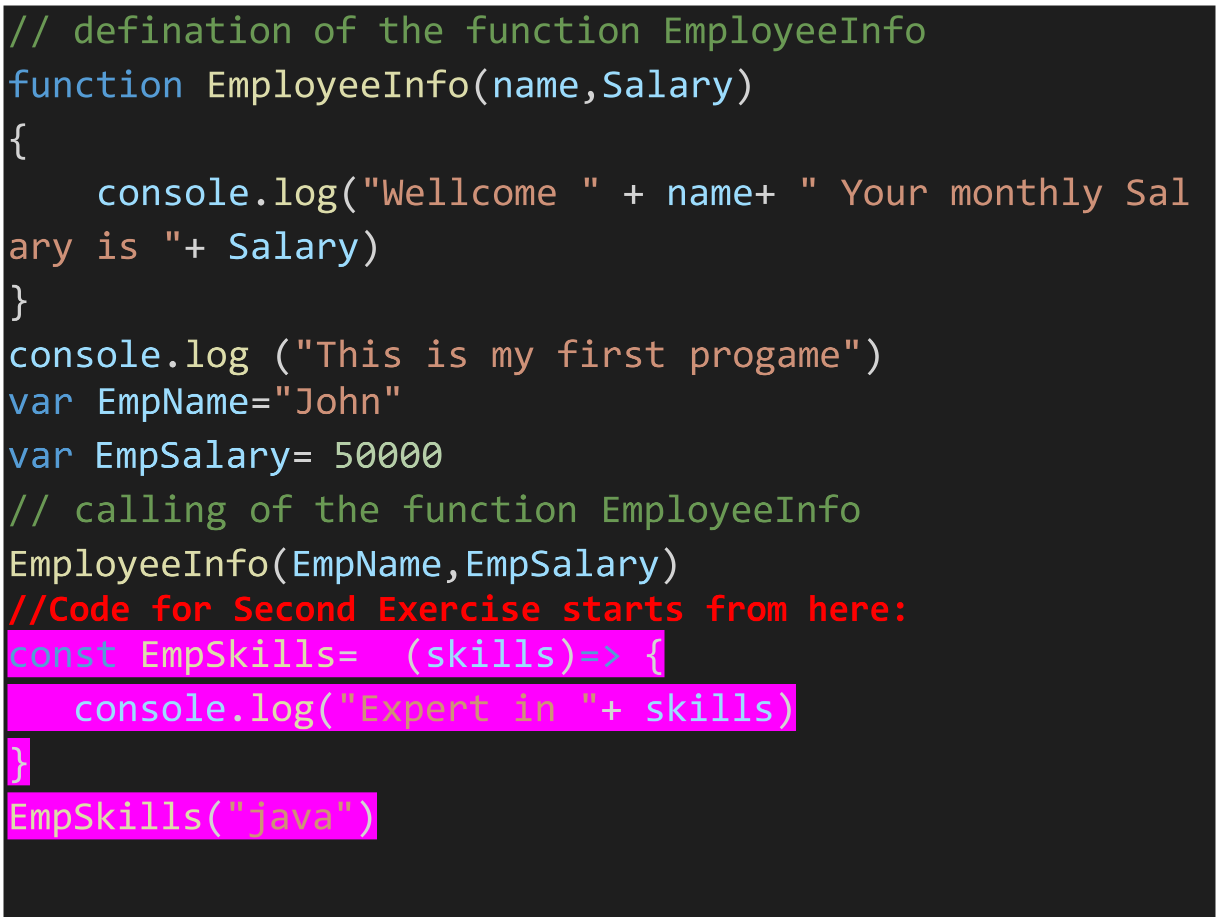
Like before the parameters are passed in (), Note function keyword is NOT used .

In your index.js file that you created for exercise 1, please Add the following code :

```
            const EmpSkills=  (skills)=> {
              console.log("Expert in "+ skills)
}
```

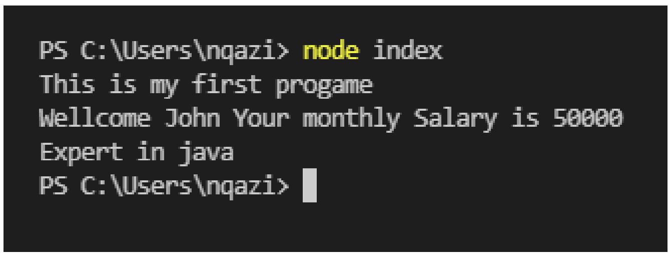
Then call this arrow function with the parameter "java" , writing following line of code :
  EmpSkills("Java") in index.js

The overall index.js should look like this :



```
// defination of the function EmployeeInfo
function EmployeeInfo(name,Salary)
{
    console.log("Wellcome " + name+ " Your monthly Sal
ary is "+ Salary)
}
console.log ("This is my first progame")
var EmpName="John"
var EmpSalary= 50000
// calling of the function EmployeeInfo
EmployeeInfo(EmpName,EmpSalary)
//Code for Second Exercise starts from here:
const EmpSkills=  (skills)=> {
   console.log("Expert in "+ skills)
}
EmpSkills("java")
```

Save the index.js and run it using node index.js you will see following output



```
PS C:\Users\nqazi> node index
This is my first progame
Wellcome John Your monthly Salary is 50000
Expert in java
PS C:\Users\nqazi>
```

Exercise3 : Creating Local Module in NodeJS

What are Java Script Modules:

A module is just a file another Java script file that holds the definition of one or more functions. You can create your own modules, and easily include them in your applications. Modules created in separate files are imported in another js file using the require key work in NodeJS

Follow these steps:

For this exercise you need three JS files.

- **StudentInfo.js (you will be creating this as separate file)**
- **Person.js (You will be creating it as separate js file)**
- **Index.js (you already have created this file Exercise 1& 2)**

Follow these steps:

Create a separate Java script file(click File →new) and save it as StudentInfo. js and write following code in this(Studentinfo.js) file:

FileName: StudentInfo.js

// arrow function with no name, this function does not take any argument vrbname = () => {}
// code starts from here:

```
const dateofBirth= "12/12/1980"

const getStudentName =    ()   =>
{
return "write your name here"
}
const getCampusName = () =>
{return ("UEL Campus ")
}
//exporting functions & variable outside the module
exports.getName=getStudentName
exports.Location=getCampusName
exports.dob=dateofBirth
// How  to export function with parameters
exports.Studentgrade=(marks)=>
        {
        if (marks>50 && marks <70) return ("B grade")
        else
            return ("A grade)")
        }
```

Step 2 : Create Another file name this file person.js.This is your another local module and type following code into it

Person.js

when you have a module that exports just one thing, it's more common to use **module.exports**. Also, observe the use of class keyword to create classes in the Java script. Constructor is another keyword used in this class. By definition, the constructor is the function that is activated when an instance of the class object is created. Write the following code in Person.js

```
class student {
    constructor(name, age, email) {
      this.name = name;
      this.age = age;
      this.email = email;
    }

    getPersonInfo() {
      return
        Name: this.name
        Age: this.age
        Email: this.email

    }
  }

  module.exports = student;
```

Step 3:Now use the functions and object defined in these two files i.e. (StudentInfo.JS, Person.JS**) into file index.js .**

we use "require" keyword to import a module in nodejs JS.

notice that in the require statement, the module name(which is simply the name of the js file where this module is defined) is prefixed with ./, as it's a local file. Also note that there's no need to add the file extension. In our case the name of modules are StudenInfo and Person as these are the names we have given the two js files.

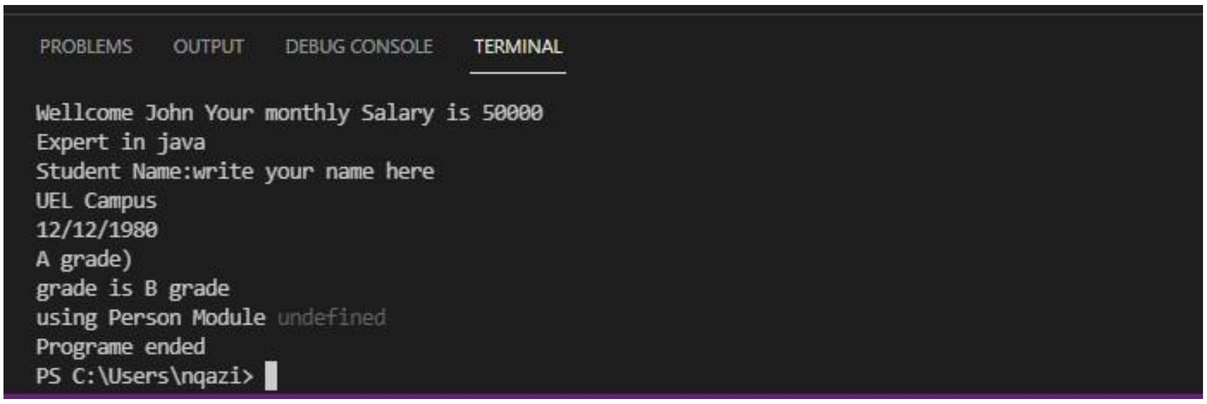
The update index.js is given below with the added code is highlighted with blue background.

index.js

```
// defination of the function EmployeeInfo
function EmployeeInfo(name,Salary)
{
    console.log("Wellcome " + name+ " Your monthly Sal
ary is "+ Salary)
}

const EmpSkills=  (skills)=> {
   console.log("Expert in "+ skills)
}
console.log ("This is my first progame")

var EmpName="John"
var EmpSalary= 50000
// calling of the function EmployeeInfo
EmployeeInfo(EmpName,EmpSalary)
EmpSkills("java")

const student= require('./StudentInfo')
const person = require('./Person')
// because getName is the function  so we  use ()
console.log("Student Name:" +student.getName())
console.log(student.Location())
console.log(student.dob)
// because dob is a variable so we do nt  use ()
console.log(student.Studentgrade())
console.log("grade is "+student.Studentgrade(55) )
// creating new Person
```

```
person1= new person("Jim","USA","myemail@gmail.com")
console.log("using Person Module",person1.getPersonInfo())
console.log("Programe ended")
```

Step 4 :Now Run it you should see following output in the console



```
Wellcome John Your monthly Salary is 50000
Expert in java
Student Name:write your name here
UEL Campus
12/12/1980
A grade)
grade is B grade
using Person Module undefined
Programe ended
PS C:\Users\nqazi>
```

Exercise 4: NodeJS core Modules

Core module used : os
Objectives: to display the laptop information on the terminal using os Module.
NodeJS functions used :require (all small letters)
Methods of core Module os used are :

Core Module used	Method of the module used	Any other Module used for this method	Syntax used
os	tmpdir	No	os.tmpdir()
os	platform	No	os.platform()
os	release	No	os.release()
os	uptime	No	os.uptime()
os	totalmem	No	os.totalmem()
os	freemem	No	os.freemem()
os	cpus	util	util.inspect(os.cpus())
os	networkInterfaces	util	util.inspect(os.networkInterfaces())

Note :console.log is used to display the text message on screen. Also it is case sensitive and all letters are small letters
A text message is always enclosed with two “” for example
Console.log(“hello”) will display a hello on screen. If you want to concatenate a textmessage with the value of variable or a function then you use a “+” Sign. For example in third line of the code below console.log("temporary directory"+ os.tmpdir()) the text message is “temporary directory ” and after + sign tmpdir() is method that will return the

name of the temporary directory and will concatenate it with the text message. All set to start your first code. It will use os core module from the nodejs and will display information about your laptop such as userinfo,hostname , free memory operating system etc.

Add following lines in your index.js files and observe the output in terminal window

```
os=require("os")
const util=require('util')
console.log("temporary directory"+ os.tmpdir() )
console.log("hostname: "+ os.hostname())
console.log("OS : " + os.platform() +"release:"+ os.release())
console.log("Uptime"+ (os.uptime())/3600 +" hours")
console.log("userInfo" + util.inspect(os.userInfo()))
console.log("Memory "+ os.totalmem()/1000000000 + "Giga byte")
console.log(" free: "+os.freemem()/1000000000 + "Giga byte")
console.log("CPU "+ util.inspect(os.cpus()))
console.log("Network"+ util.inspect(os.networkInterfaces()))
console.log("programe end")
```

node index.js to run the program

For your portfolio, write a report covering today's lab focusing on the following :

Submit the code for the completed Exercises i.e.:

i) **Index.js**
ii) **Person.js**
iii) **Employeeinfo.js**
iv) **Exercise 4.js**