# Department of Computing

**CS-213: Advanced Programming**

**Class: BSCS 7AB**

# Lab 4: Introduction to Node.js

**Date: 26rd September, 2019**

**Time: 10:00-01:00pm & 02:00-05:00pm**

**Name:Tatheer Zahra**

**Reg#211972**

**Lab Tasks**

**Task 1:** Download Node.js from the official Node.js web site: [https://nodejs.org](https://nodejs.org/).

**Task 2:**

**Code:**

var http = require('http');

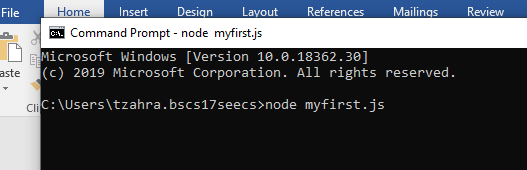
http.createServer(function (req, res) {

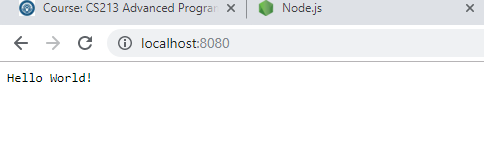
res.writeHead(200, {'Content-Type': 'text/plain'});

res.end('Hello World!');

}).listen(8080);

**Output:**





**Task 3:** Create a module that returns the current date and time. Save the code in a file called "myfirstmodule.js".

**Code:**

exports.myDateTime = function (){

return Date();

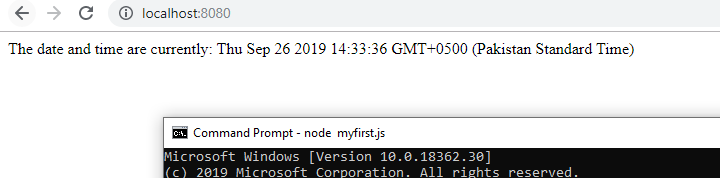
};

**Task 4:** Use the module "myfirstmodule" of date and time in a Node.js file.

**Code:**

var http = require('http');  
**var dt = require('./myfirstmodule');**  
http.createServer(function (req, res) {  
  res.writeHead(200, {'Content-Type': 'text/html'});  
  res.write("The date and time are currently: " + **dt.myDateTime()**);  
  res.end();  
}).listen(8080);

**Output:**



**Node.js as a Web Server**

**Task 5:** The HTTP module can create an HTTP server that listens to server ports and gives a response back to the client. Use the createServer() method to create an HTTP server.

**Code:**

var http = require('http');

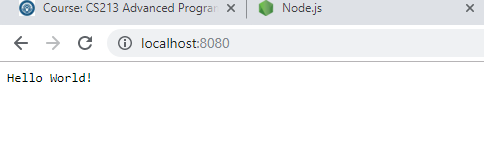
http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/plain'});

res.end('Hello World!');

}).listen(8080);

**Output:**



**Task 6:** Add an HTTP Header

**Code:**

var http = require('http');

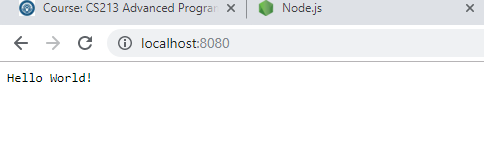
http.createServer(function (req, res) {

res.writeHead(200, {'Content-Type': 'text/plain'});

res.end('Hello World!');

}).listen(8080);

**Output:**



**Task 7:** Create a Node.js file that reads the HTML file, and return the content.

**Code:**

**JS File:**

var http = require('http');

var fs = require('fs');

http.createServer(function (req, res) {

fs.readFile('demo.html', function(err, data) {

res.writeHead(200, {'Content-Type': 'text/html'});

res.write(data);

res.end();});

}).listen(8080);

**Html file:**

<!DOCTYPE html>

<html>

<head>

<title>Page Title</title>

</head>

<body>

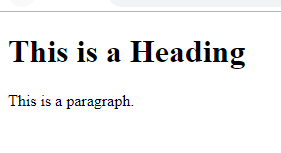
<h1>This is a Heading</h1>

<p>This is a paragraph.</p>

</body>

</html>

**Output:**



**Task 8:** Create a new file using

* appendFile() method

**Code:**

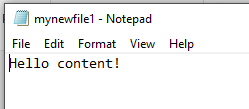
var fs = require('fs');

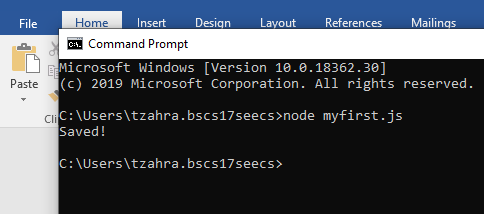
fs.appendFile('mynewfile1.txt', 'Hello content!', function (err) {

 if (err) throw err;

console.log('Saved!');});

**Output:**





* open() method

**Code:**

var fs = require('fs');

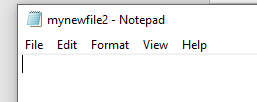
fs.open('mynewfile2.txt', 'w', function (err, file) {

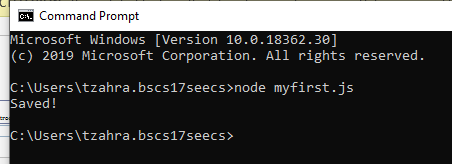
if (err) throw err;

console.log('Saved!');

});

**Output:**





* writeFile() method

**Code:**

var fs = require('fs');

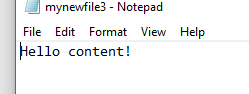
fs.writeFile('mynewfile3.txt', 'Hello content!', function (err) {

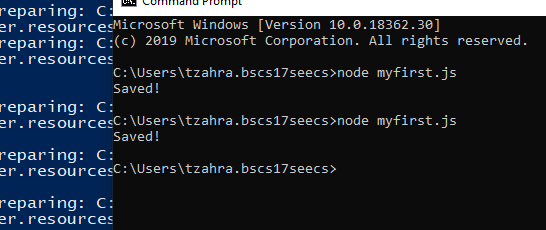
if (err) throw err;

console.log('Saved!');

});

**Output:**





**Task 9:** Append "This is my text." to the end of the file "mynewfile1.txt".

**Code:**

var fs = require('fs');

fs.appendFile('mynewfile1.txt', ' This is my text.', function (err) {

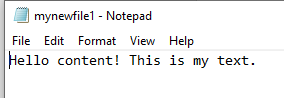
if (err) throw err;

console.log('Updated!');

});

**Output:**





**Task 10:** Replace the content of the file "mynewfile3.txt".

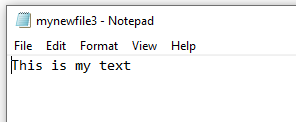
var fs = require('fs');

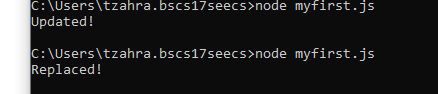
fs.writeFile('mynewfile3.txt', 'This is my text', function (err) {

if (err) throw err;

console.log('Replaced!');

});





**Task11:** Delete "mynewfile2.txt".

Code:

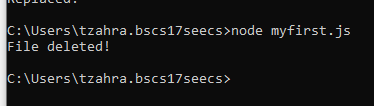
var fs = require('fs');

fs.unlink('mynewfile2.txt', function (err) {

if (err) throw err;

console.log('File deleted!');

});



**Task 12:** Rename "mynewfile1.txt" to "myrenamedfile.txt".

Code:

var fs = require('fs');

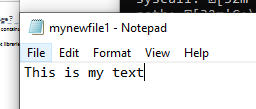
fs.rename('mynewfile1.txt', 'myrenamedfile.txt', function (err) {

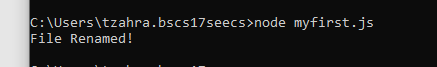
if (err) throw err;

console.log('File Renamed!');

});

**Before:**





**After:**

