**Unit 2**

**Q.1) Create an application to demonstrate Node.js Events.**

var events = require("events");

var em = new events.EventEmitter();

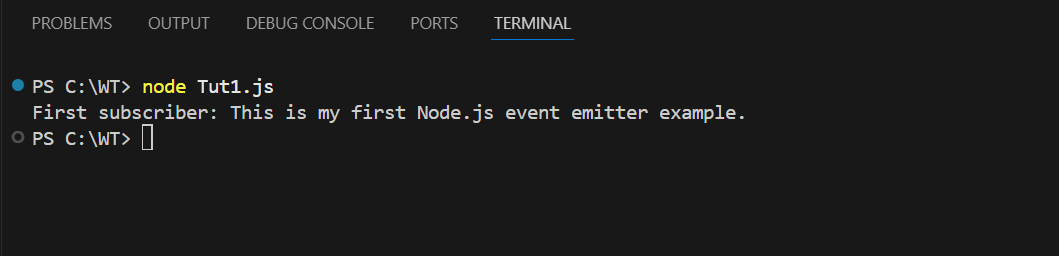
em.on("FirstEvent", function (data) {

  console.log("First subscriber: " + data);

});

em.emit("FirstEvent", "This is my first Node.js event emitter example.");

Output :



**Custom event :**

const events = require("events");

const eventEmitter = new events.EventEmitter();

eventEmitter.on("connection", handleConnectionEvent);

eventEmitter.emit("connection");

eventEmitter.emit("connection");

eventEmitter.emit("connection");

eventEmitter.emit("connection");

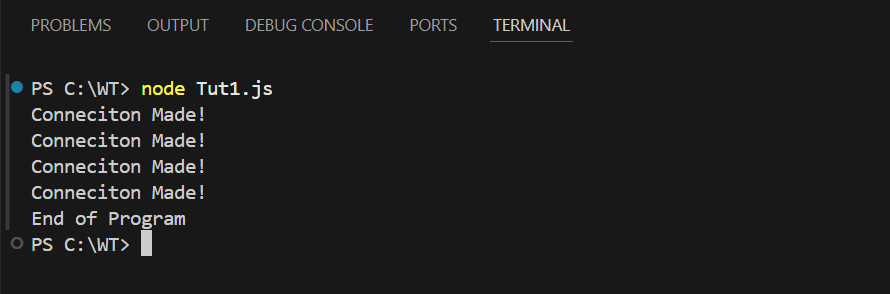
function handleConnectionEvent() {

  console.log("Conneciton Made!");

}

console.log("End of Program");

Output :



var events = require("events");

var eventEmitter = new events.EventEmitter();

var connectHandler = function connected() {

  console.log("connection successful.");

  eventEmitter.emit("data\_recieved");

};

eventEmitter.addListener("connection", connectHandler);

eventEmitter.addListener("data\_recieved", function () {

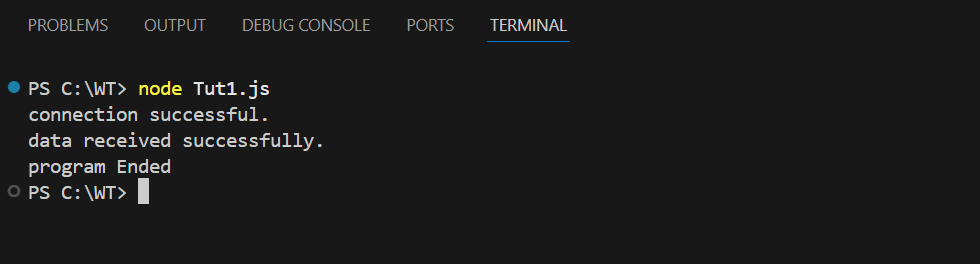
  console.log("data received successfully.");

});

eventEmitter.emit("connection");

console.log("program Ended");

Output :



**Q.2) Implement all the methods of EventEmitter class.**

const events = require("events");

const eventEmitter = new events.EventEmitter();

function listner1() {

  console.log("Event recevied by Listner 1");

}

function listner2() {

  console.log("Event recevied by Listner 2");

}

eventEmitter.addListener("write", listner1);

eventEmitter.on("write", listner2);

eventEmitter.emit("write");

console.log(eventEmitter.listenerCount("write"));

eventEmitter.removeListener("write", listner1);

console.log("Listener 1 is removed");

eventEmitter.emit("write");

console.log(eventEmitter.listenerCount("write"));

console.log("Program Ended");

Output :



**Implement Event Emitter Patterns**

1. using return value of function

var emitter = require("events").EventEmitter;

function LoopProcessor(num) {

var e = new emitter();

setTimeout(function () {

for (var i = 1; i <= num; i++) {

e.emit("BeforeProcess", i);

console.log("Processing number:" + i);

e.emit("AfterProcess", i);

}

}, 2000);

return e;

}

var lp = LoopProcessor(3);

lp.on("BeforeProcess", function (data) {

console.log("About to start the process for " + data);

});

lp.on("AfterProcess", function (data) {

console.log("completed processing " + data);

});

Output :



1. Extend Event emitter class(using Util module).

var emitter = require("events").EventEmitter;

var util = require("util");

function LoopProcessor(num) {

var me = this;

setTimeout(function () {

for (var i = 1; i <= num; i++) {

me.emit("BeforeProcess", i);

console.log("Processing Number: " + i);

me.emit("AfterProcess", i);

}

}, 2000);

return this;

}

util.inherits(LoopProcessor, emitter);

var lp = new LoopProcessor(3);

lp.on("BeforeProcess", function (data) {

console.log("About to start the process for" + data);

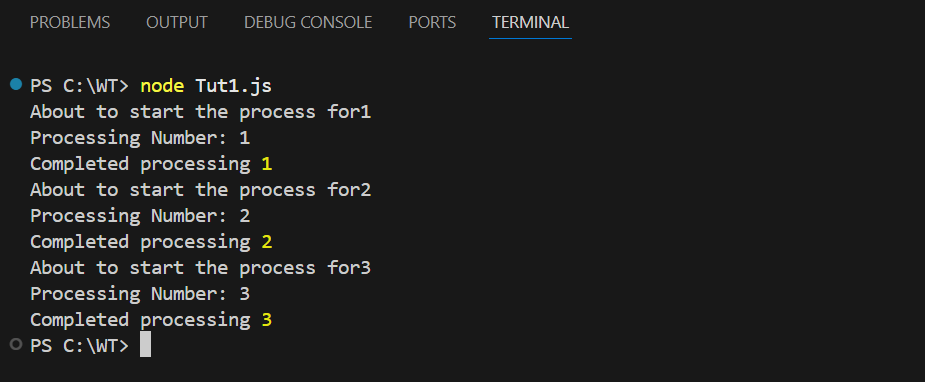
});

lp.on("AfterProcess", function (data) {

console.log("Completed processing", data);

});

Output :



**Q.3) Create an application to demonstrate Node.js Functions.**

**Callback functions**

/\* What is Call back function

A callback is a function passed as an argument to another function.

\*/

//callback function - Anonymous Function

const message=function(){

console.log("Hi I am Bruce Wayne ");

}

setTimeout(message,3000);

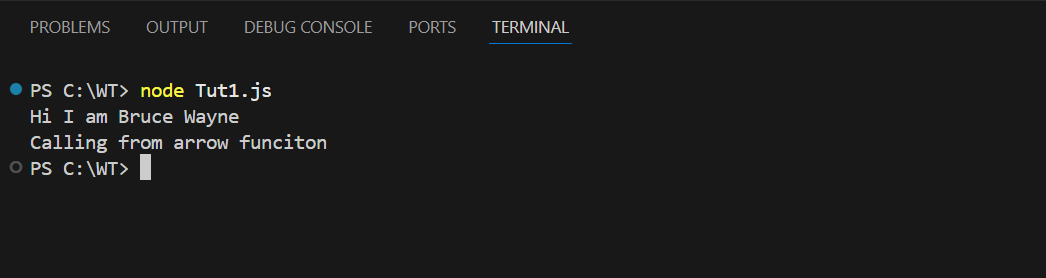
//callback back as an Arrow function

setTimeout(()=>{

console.log("Calling from arrow funciton");

},3000);’

Output :



function displayresult(some)

{

console.log(some);

}

function calculate(x,y,mycallback)

{

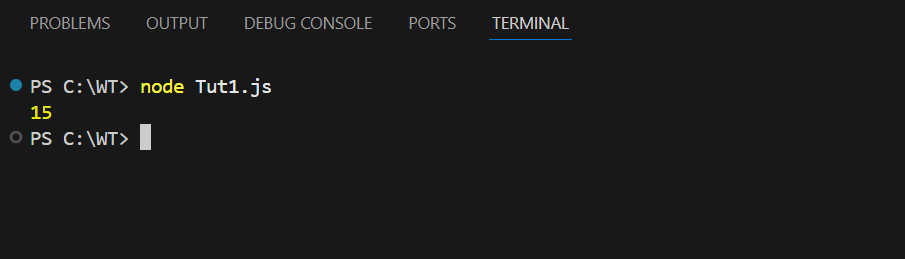
let sum=x+y;

mycallback(sum);

}

calculate(5,10,displayresult);

Ouput :



Standard function :

function myfun(num1,num2)

{

console.log(num1+num2);

console.log(num1-num2);

console.log(num1\*num2);

console.log(num1/num2);

}

myfun(12,3);

Output :

