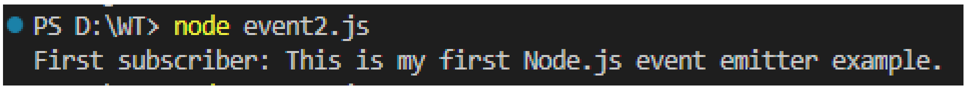
Practical No : 2

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:



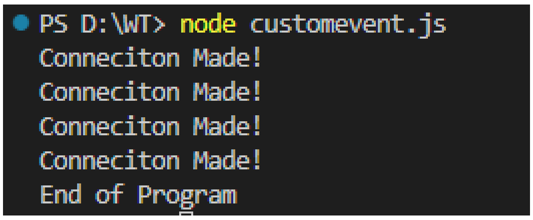
```
PS D:\WT> node event2.js
  First subscriber: This is my first Node.js event emitter example.
```

q

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:

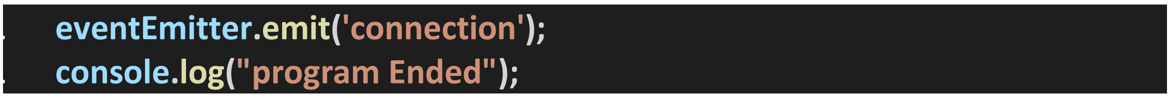


```
PS D:\WT> node customevent.js
  Conneciton Made!
  Conneciton Made!
  Conneciton Made!
  Conneciton Made!
  End of Program
```

Implement all Methods of EventEmitter class.

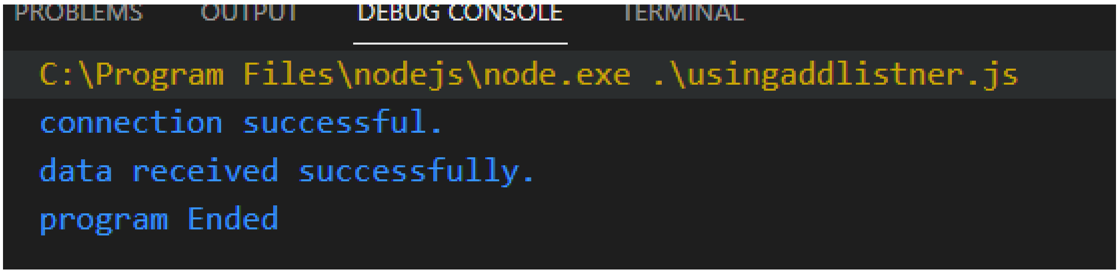
I. **using addlistner method**

```
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.');
   });
```



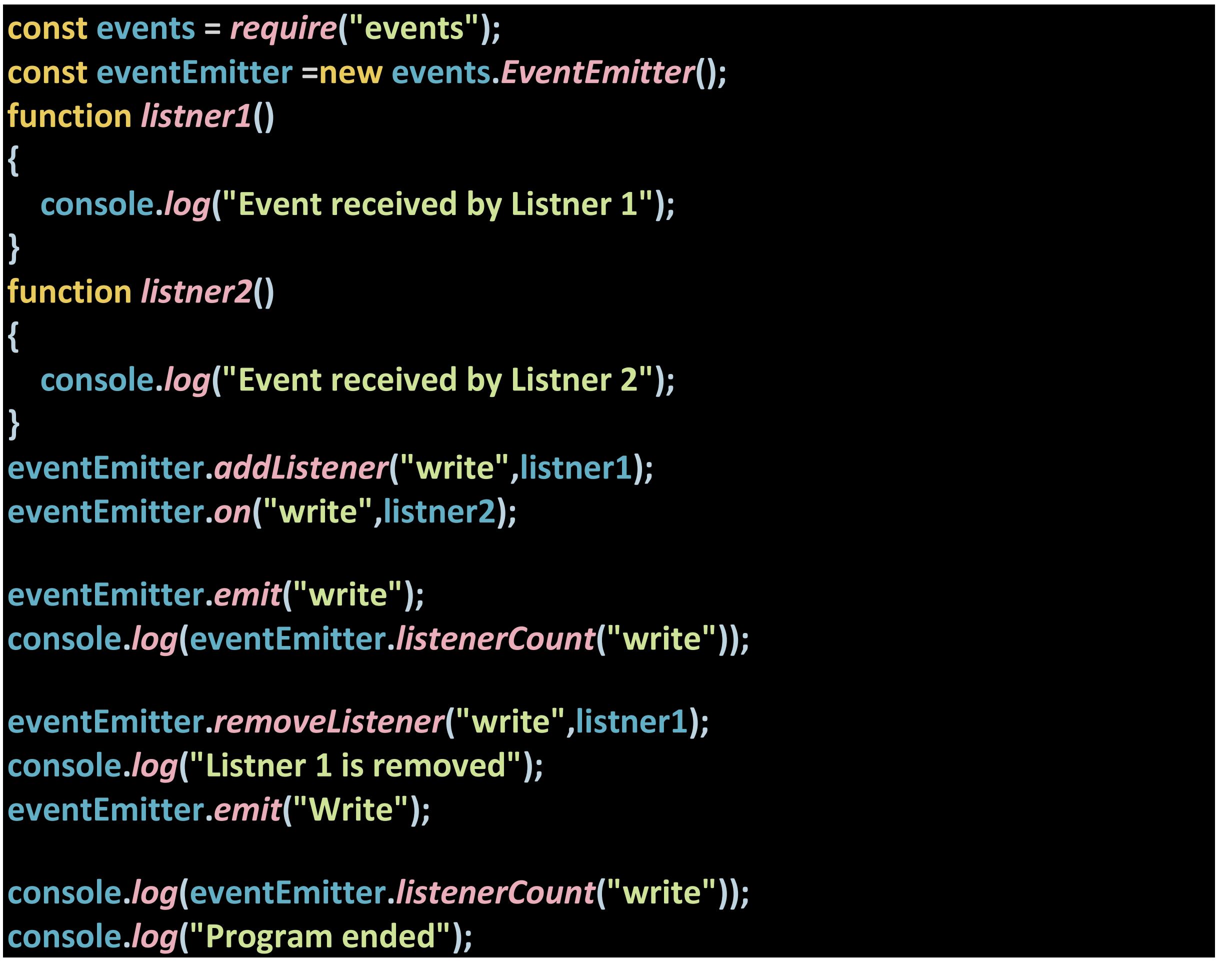
```
    eventEmitter.emit('connection');
    console.log("program Ended");
```

Output:



```
  C:\Program Files\nodejs\node.exe .\usingaddlistner.js
  connection successful.
  data received successfully.
  program Ended
```

II. **Remove Listner**



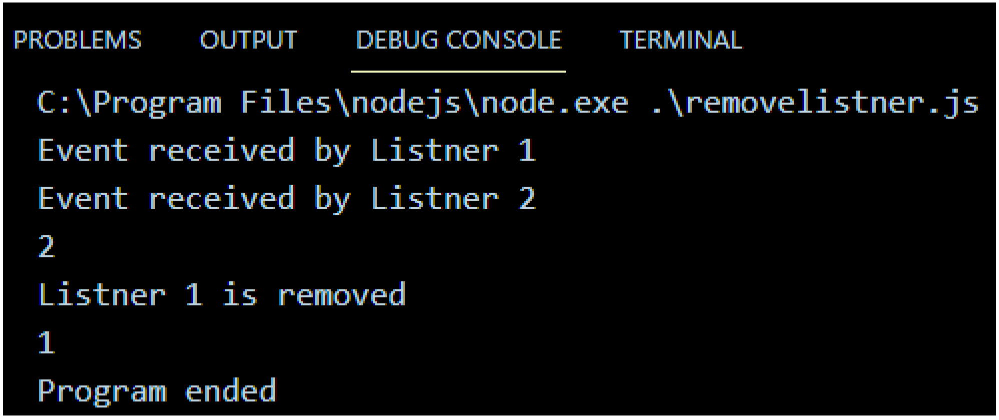
```
const events = require("events");
const eventEmitter =new events.EventEmitter();
function listner1()
{
   console.log("Event received by Listner 1");
}
function listner2()
{
   console.log("Event received by Listner 2");
}
eventEmitter.addListener("write",listner1);
eventEmitter.on("write",listner2);

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

eventEmitter.removeListener("write",listner1);
console.log("Listner 1 is removed");
eventEmitter.emit("Write");

console.log(eventEmitter.listenerCount("write"));
console.log("Program ended");
```

Output:



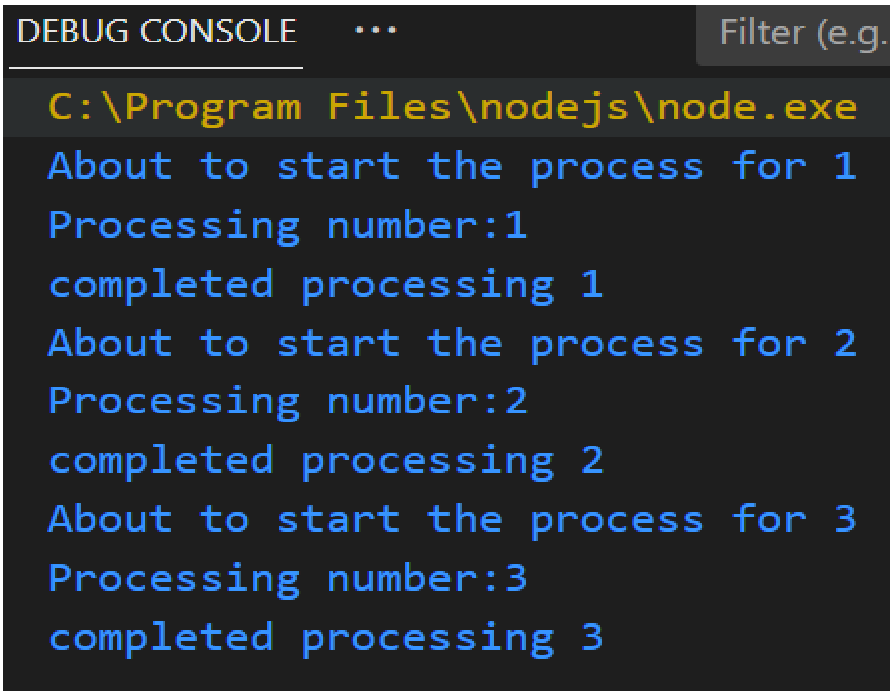
```
  C:\Program Files\nodejs\node.exe .\removelistner.js
  Event received by Listner 1
  Event received by Listner 2
  2
  Listner 1 is removed
  1
  Program ended
```

Implement Event Emitter Patterns

a)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);
    });
```

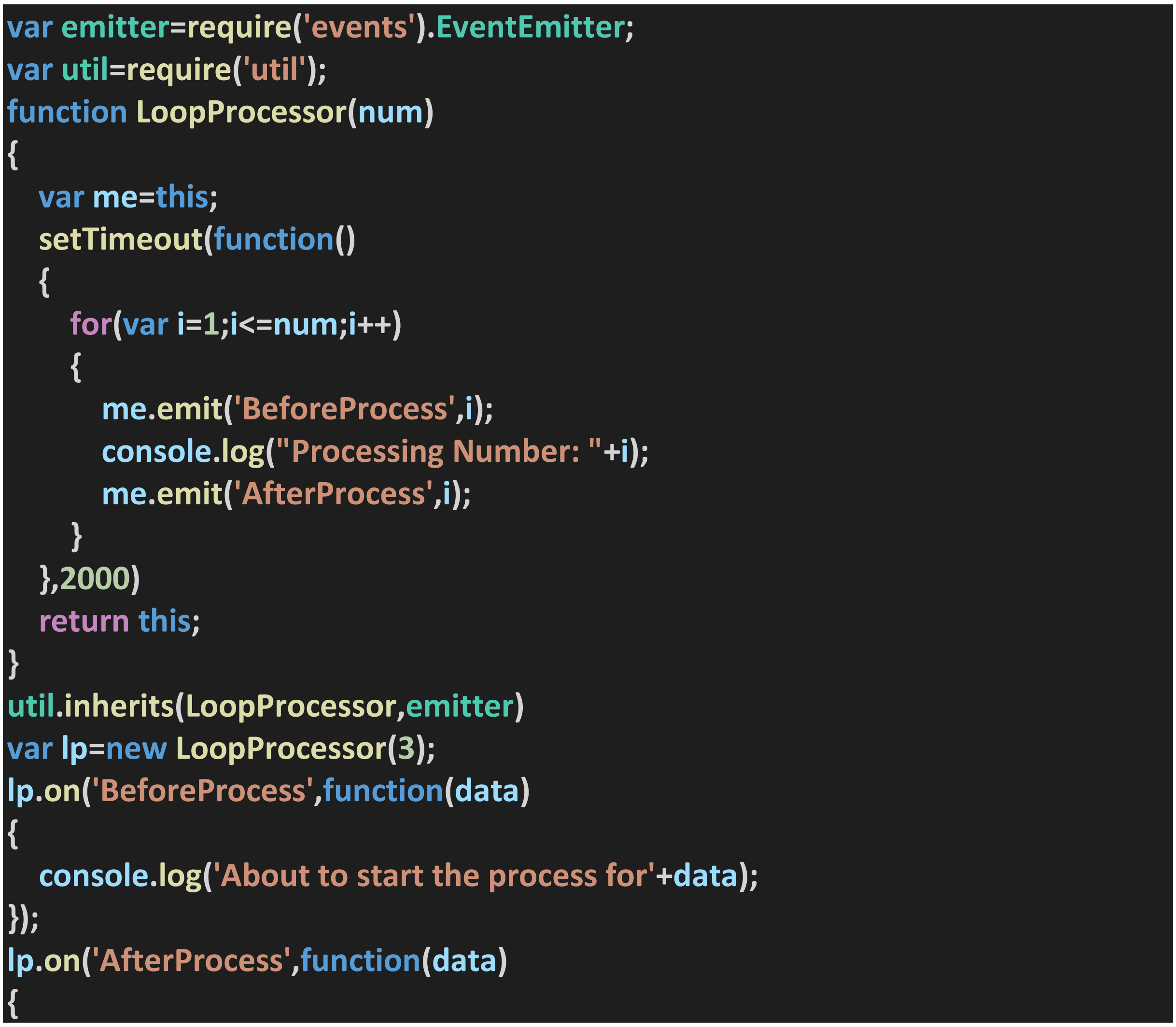
Output:



```
C:\Program Files\nodejs\node.exe
About to start the process for 1
Processing number:1
completed processing 1
About to start the process for 2
Processing number:2
completed processing 2
About to start the process for 3
Processing number:3
completed processing 3
```

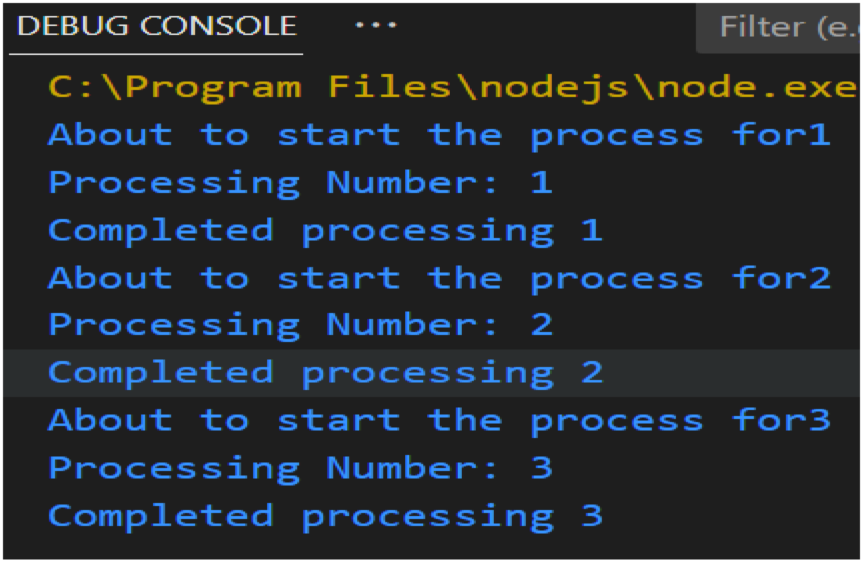
b)Extend Event emitter class(using Util module)

Extend Event Emitter Class



```
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:



```
C:\Program Files\nodejs\node.exe
About to start the process for1
Processing Number:  1
Completed processing 1
About to start the process for2
Processing Number:  2
Completed processing 2
About to start the process for3
Processing Number:  3
Completed processing 3
```