

Bài tập 14.1 Viết lại các đoạn mã trong bài học.

```
server đang chạy trên cổng 9000  
[nodemon] restarting due to changes...  
[nodemon] starting `node index.js`  
Đang chờ  
server đang chạy trên cổng 9000  
Hết giờ học rồi, về thôi!!!!!!  
□
```

Bài tập 14.2 Sử dụng module `events` của Nodejs để lập trình hướng sự kiện. Viết một chương trình đơn giản như sau:

- Khai báo sự kiện `vaoLop` (vào lớp), với hàm xử lý đi kèm là `xuLyVaoLop`
- Sử dụng hàm `setTimeout()` để hẹn giờ vào lớp là 5 giây
- Hết 5 giây, web server sẽ thông báo “Đã đến giờ học” tới cửa sổ console của web server
- Viết bằng 2 cách: hàm thông thường và hàm mũi tên.

```
23 const events = require('events');
24 const EventEmitter = new events.EventEmitter();
25 /* Viết hàm thông thường
26 eventEmitter.on('vaolop', xuLyVaoLop);
27
28 function xuLyVaoLop(tb){
29 |   console.log(tb);
30 | }
31
32 setTimeout(()=>{
33 |   eventEmitter.emit('vaolop', 'Đã đến giờ vào lớp')
34 | }, 5000);
35 */
36
37 //hàm mũi tên
38 eventEmitter.on('vaolop', (tb)=>{
39 |   console.log(tb)
40 | });
41
42 setTimeout(()=>{
43 |   eventEmitter.emit('vaolop', 'Đã đến giờ vào lớp')
44 | }, 5000);
45
46 console.log("Đang chờ");
47
48
49 app.listen(port, () => {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Đã đến giờ vào lớp
[nodemon] restarting due to changes...
[nodemon] starting `node index.js`
Đang chờ
server đang chạy trên cổng 9000
Đã đến giờ vào lớp
```

Câu hỏi 14.3 Mô hình lập trình hướng sự kiện trong Node.js là gì? Phát biểu nào sau đây **không đúng**?

- A. Event Emitter liên tục kiểm tra hàng đợi sự kiện và thực thi các hàm callback khi có sự kiện xảy ra.
- B. Event là các hành động hoặc sự thay đổi trạng thái xảy ra trong ứng dụng.
- C. Event Loop liên tục kiểm tra hàng đợi sự kiện và thực thi các hàm callback khi có sự kiện xảy ra.
- D. Event Handler là các hàm được gọi khi một sự kiện xảy ra.