Class 13

Limitations to callbacks

- Fire only once.
- Recursion can be used to get multiple fires, but the code can get messy.
- Callback APIs require a callback

```
const readline = require("readline");
//const rl = readline.createInterface(process.stdin, process.stdout);

// rl.question(">", (err, data) => {
//   console.log(data);
// });

// console.log("Non blocking");

// rl.on("line", (data) => {
//   console.log(`Recieved ${data}`);
// });

// console.log("Enter any input and press enter to fire line event");

const rl = readline.createInterface({ input: process.stdin });
const print = (input) => {
  console.log(`Received: ${input}`);
};

rl.on("line", print);
rl.on("line", print);

console.log(
  "Event Listeners will each indepently fire when an event is emitted"
);

/*
Observer pattern
    - Subject (EventEmitter, Resource, Observable)
        emit(eventName, [data1, data2,...]);
            -eventName : string
            -data1, data2, ... : data passed to observers
    - Observer (EventListener)
        on(eventName, callback);
            -eventName : string that matches emit
            -callback : function to fire after each emit
                -data1, data2, forwarded from emitter

*/

const EventEmitter = require;
```