

### Practice Problem Set 3

Complete after practice\_12.md.

Q1	Final 201712 P20	Listener vs. emitter	___ / 8 points
Q2	Final 201703 P10	Listener vs. emitter	___ / 12 points
Q3	Midterm 201703 P5	Event listener	___ / 12 points
Q4	Midterm 201606 P6	Event listener	___ / 12 points
Q5	Midterm 201603 P5	Event listener	___ / 9 points
Q6	Midterm 201712 P4	Event listener	___ / 12 points
Q7	Midterm 201806 P1	Event listener	___ / 10 points
		Total	___ / 75 points

**Problem #20 (8 points)**

The code fragment below uses `async` and `events` module. Write out the `console.log` outputs in the order in which they will be printed for this code fragment.

```
var async = require('async');
var events = require('events');

var emitterOne = new events.EventEmitter();
var emitterTwo = new events.EventEmitter();

function sumWithOne(num, callback) {
    sum = 1 + num;
    emitterOne.emit('event 1');
    callback();
}

emitterOne.on('event 1', function() {
    console.log('A');
});
emitterTwo.on('event 1', function() {
    console.log('B');
});
emitterOne.on('event 2', function() {
    console.log('D');
});
emitterOne.on('event 2', function() {
    console.log('E');
});

console.log('Start');
async.each([1, 2], sumWithOne, function(err) {
    console.log('Async done');
    emitterOne.emit('event 2');
    emitterTwo.emit('event 2');
});
console.log('End');
```

**Problem #10 (12 points)**

The following Node.js program uses the Node `events` and `fs` modules to read an input file and print some lines to the console log.

```
var events = require('events');
var myEmitter = new events.EventEmitter();

var fs = require('fs');

myEmitter.on('A', function () {
  console.log('A');
});

myEmitter.on('D', function () {
  console.log('D');
});

var readableStreamEvent = fs.createReadStream('./inputFile');

readableStreamEvent.on('data', function (fileData) {
  console.log('B');
  myEmitter.emit('A');
});

myEmitter.emit('D');

readableStreamEvent.on('finish', function () {
  console.log('C');
  myEmitter.emit('A');
});
```

**Answer questions on the following page.....**

... problem #10 continued from previous page.

1. Assuming that `inputFile` contains exactly one character, list the letters that will be printed to the console log in the order in which they will be printed.
2. Assuming you have control over the contents of `inputFile`, is it possible to see A printed more than twice? Why or why not? Briefly justify your answer.
3. Assuming you have control over the contents of `inputFile`, is it possible to see A printed less than twice? Why or why not? Briefly justify your answer.

## Problem #5 (12 points)

Given the following simple HTML document that contains the word "Div" and some JavaScript that registers event handlers on it.

```
<html>
  <body id="body">
    <div id="div">Div</div>
  </body>
</html>
```

```
document.getElementById("body").addEventListener("click",
  function (e) {console.log('Body-true', e.target===e.currentTarget)},
  true);
document.getElementById("body").addEventListener("click",
  function (e) {console.log('Body-false',
e.target===e.currentTarget)},
  false);
document.getElementById("div").addEventListener("click",
  function (e) { console.log('Div-true',
e.target===e.currentTarget)},
  true);
document.getElementById("div").addEventListener("click",
  function (e) {
console.log('Div-false',e.target===e.currentTarget)},
  false);
(Hint: The documentation for addEventListener names its last parameter useCapture.)
```

(a) Describe what would be printed to the `console.log` if you clicked on the word "Div".

(b) Describe what would be printed to the console log if you clicked on the page but not on or near the word "Div".

## Problem #3 (12 points)

(A)

Given an HTML page with a button element, with id of "button" and the following JavaScript:

```
console.log("A");

setTimeout( function() {
    console.log("B");
}, 1000);

console.log("C");

setTimeout( function() {
    console.log("D");
}, 1500);

document.getElementById("button").onclick = function() {
    console.log("E");
    console.log("F");
}
```

Assuming the JavaScript is executed when the window is loaded, and a user clicks on the element with id "button" at some point, circle the following console outputs which are **possible** (there may be more than one).

A	A	A	A	A
B	C	C	C	C
C	B	B	E	E
D	E	E	B	F
E	F	D	F	B
F	D	F	D	D

(B)

If you had both a capture phase listener and a bubble phase listener for a click event on a DOM element, which would expect to run first if you clicked on the element? Briefly justify your answer.

How about if you click on a child element of the element with the listener registered? Briefly justify your answer.

## Problem #5 (9 points)

Consider the following fragments of HTML and JavaScript:

----- HTML -----

```
<div id="a">A
  <div id="b">B
    <div id="d">D</div>
  </div>
  <div id="c">C</div>
</div>
```

----- JavaScript -----

```
function callback(event) {
  console.log(event.target.id + ' ' + event.currentTarget.id)
}
document.getElementById('a').addEventListener("click", callback, false);
document.getElementById('b').addEventListener("click", callback, false);
document.getElementById('c').addEventListener("click", callback, false);
document.getElementById('d').addEventListener("click", callback, false);
```

- I. What would be printed out if the user clicked on the letter 'D'?
  
  
  
  
  
  
  
  
  
  
- II. If the order of the calls to `addEventListener` were reversed so we instead registered the listeners in the div id order of 'd', 'c', 'b', 'a', what would be printed out if the user clicked on the letter 'D'?
  
  
  
  
  
  
  
  
  
  
- III. If we change the JavaScript so that the third argument to each `addEventListener()` call is `true` (i.e. register them as capture rather than bubble listeners) and keep the original order from Part I, what would be printed out if we clicked on the letter 'C'?

## Problem #4 (12 points)

Consider the following HTML and JavaScript program:

```
<!doctype html>
<html>
  <head>
  </head>
  <body>
    <div id="container">
      <button id="button">Click me!</button>
    </div>
  </body>
  <script>
    'use strict';
    function printWindow(event) {
      console.log('Window says hello!');
    }

    function printTarget(event) {
      console.log('Target says hello!');
    }

    function printContainer(event) {
      console.log('Container says hello!');
    }

    var capture = true;
    var button = document.getElementById('button');
    var container = document.getElementById('container');

    window.addEventListener('click', printWindow);
    button.addEventListener('click', printTarget, capture);
    container.addEventListener('click',
printContainer,capture);

  </script>
</html>
```

The page loads and the user waits a few seconds before clicking the button that says "Click me!". What's the console output?



## Problem #1 (10 points)

The following HTML document is loaded into a browser window:

```
<html>
<body>
  <div id="D">D
    <div id="A">A</div>
    <div id="B">B
      <div id="C">C</div>
    </div>
  </div>
  <script type="text/javascript">
    function callback(event) {
      console.log(event.currentTarget.id);
    }
    document.getElementById("D").addEventListener("click", callback, true);
    document.getElementById("A").addEventListener("click", callback, false);
    document.getElementById("B").addEventListener("click", callback, false);
    document.getElementById("C").addEventListener("click", callback, true);
  </script>
</body>
</html>
```

Hint: the last parameter to `addEventListener` is a boolean indicating capture phase (`true`) or bubble phase (`false`).

Part (a): Draw out what the browser would display for this document: