Lecture 10 Activity: Callbacks and Timers in JS

Pre-Check Q4: This problem uses setTimeout with callbacks. For the following program, write the order of statements output after the program is executed.

Program	Output (4 lines)
(see program below)	

```
(function() {
  console.log("Foo 1");
  window.addEventListener("load", init);

  function init() {
    setTimeout(testFunction, 1000);
    console.log("Foo 2");
  }

  function testFunction() {
    console.log("Foo 3");
  }

  console.log("Foo 4");
})();
```

Review:

```
let timerId = null; // stores ID of our interval timer

function init() {
  id("toggle-btn").addEventListener("click", toggleMessageInterval);
}

// 1. What does this function do?
function toggleMessageInterval() {
  if (timerId === null) {
    timerId = setInterval(sayHello, 1000);
  } else {
    clearInterval(timerId);
    timerId = null; // 2. Why is this line important?
    // 3. What happens if we swap the two lines above?
  }
}

function sayHello() {
  id("output-text").textContent += "Hello...";
}
```

Lecture 10 Activity: Callbacks and Timers in JS

Extra practice (at home): This one is a bit tricky, but is really good to practice tracing event flow with asynchronous functions like setTimeout and setInterval. Consider the following JS program:

```
(function () {
  let t1 = null;
  let t2 = null;
  let doggoCount = 0;

  window.addEventListener("load", init);

  function init() {
    t1 = setInterval(doggo, 300);
    t2 = setTimeout(dubs, 800);
  }

  function doggo() {
    doggoCount += 1;
    console.log(doggoCount + " doggo");
  }

  function dubs() {
    console.log("DUBS!");
    t1 = null;
    clearInterval(t1);
    t2 = setTimeout(dubs, 800);
  }

})();
```

Circle which of the following options would be correct as the first 8 lines of console output when the page is loaded (fewer than 8 lines indicate no more console output is possible until the program is restarted).

a	b	c	d	e
1 doggo 2 doggo DUBS! 3 doggo 4 doggo DUBS! 5 doggo 6 doggo	1 doggo 2 doggo 3 doggo DUBS!	1 doggo 2 doggo DUBS! 1 doggo 2 doggo DUBS! 1 doggo 2 doggo	1 doggo 2 doggo DUBS! 3 doggo 4 doggo 5 doggo DUBS! 6 doggo	1 doggo 2 doggo DUBS! DUBS! DUBS! DUBS! DUBS! DUBS!