**1.**

Demonstrate how to delete property from an object.

Eg. Delete message.text will delete the object’s property .text.

**2.**

What will be logged and **why**?

var id = 0;

var events = [];

events[++id] = { id: id, description: 'Graduation' };

console.log(events[0]); // ??? undefined as no value given to array.

console.log(events[1].id); // ???First it increments id – so array[1] is there and id is there. Property id, has reference to id and this has been upped to a value of 1, so that (1) is the value returned.

If events[0] this would give a type error.

**3.**

What will be logged and **why**?

var TAX = 20; function sum()

{ var total = 0;

total = arguments[0] + arguments[1] + arguments[2];

total = total + (total \* this.TAX / 100);

return total; }

var result = sum.bind(null).apply(null, [30, 70, 900]); console.log(result); // ???

It returns 1200.

.apply uses the first argument to assign to this and then the remaining arguments are in arguments[].

Bind does not call a function but returns a function.

**4.**

What will be logged and **why**?

var message = {}.prototype = [].constructor.prototype; console.log(message.length); // ???

[].constructor.prototype = [] therefore var message is assigned empty array, and empty array has length of 1 (as the length of the elements is 0.

**5.**

What will be logged and **why**?

console.log(null === null ? null : false);

Same as null===null else null is false

If (null===null){return null}else{return false}

Therefore this would return null.

**6.**

Implement reverse function:

[true, 100, 'Hello', {}, []].reverse();

can do with recursion.

Logic –pop first element to second array?

// [[], {}, 'Hello', 100, true]