

COMP 333
Fall 2019

Prototype-based Inheritance

1.) Consider the JavaScript code below:

```
function Base() {}
function Sub1() {}
function Sub2() {}

// <<some additional code>>

(new Base()).method();           // prints "base"
(new Sub1()).method();           // prints "sub1"
(new Sub2()).method();           // prints "base"
console.log(Sub1 instanceof Base); // prints "true"
console.log(Sub2 instanceof Base); // prints "true"
```

Code is elided where <<some additional code>> is. Write what this elided code must be below.

2.) Consider the JavaScript code below. What is the output of this code?

```
function AddThis(x) { this.x = x; }
AddThis.prototype.add = function (y) { return this.x + y; }

let withOne = new AddThis(1);
let withFive = new AddThis(5);
console.log(withOne.add(1));
console.log(withOne.add(2));
```

3.) Write JavaScript code which will add a `sub` method to all instances of `AddThis`, where `sub` should subtract `this.x` from its parameter and return the result.

4.) Write JavaScript code which will add a `mul` method to **newly-created** instances of `AddThis`, where `mul` should multiply `this.x` with its parameter and return the result. **Newly-created** `AddThis` instances should have the same `add` and `sub` methods as before, without repeating their definitions. Existing instances of `AddThis` should **not** have a `mul` method.