SWA1 Exercises 1

Exercise 1.1 – Installation

a) Install node.js from https://nodejs.org/en/

b) Run the program below. What does it do?

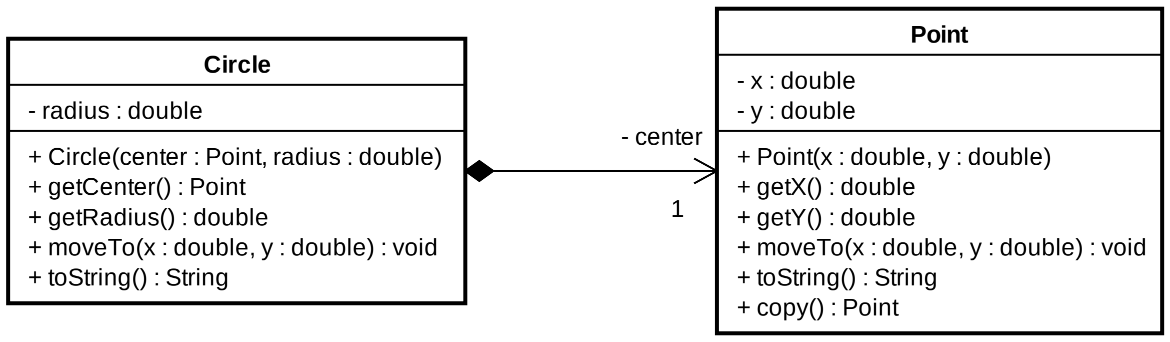
```
const chars = {
  '1': 'e',
  '8': 'r',
  '11': '!',
  '4': 'o',
  '0': 'H',
  '10': 'd',
  '6': 'W',
  '9': 'l',
  '2': 'l',
  '7': 'o',
  '3': 'l',
  length: 12
}

let msg = ''
for(let i = 0; i < chars.length; i++) {
  if (chars[i])
      msg = msg + chars[i]
  else
      msg = msg + ' '
}

console.log(msg)
```

Exercise 1.2 – Implement class model

a) The class diagram below is a fine Java class diagram, but it is not how we program object-oriented in JavaScript. Implement the model in JavaScript using factory functions.



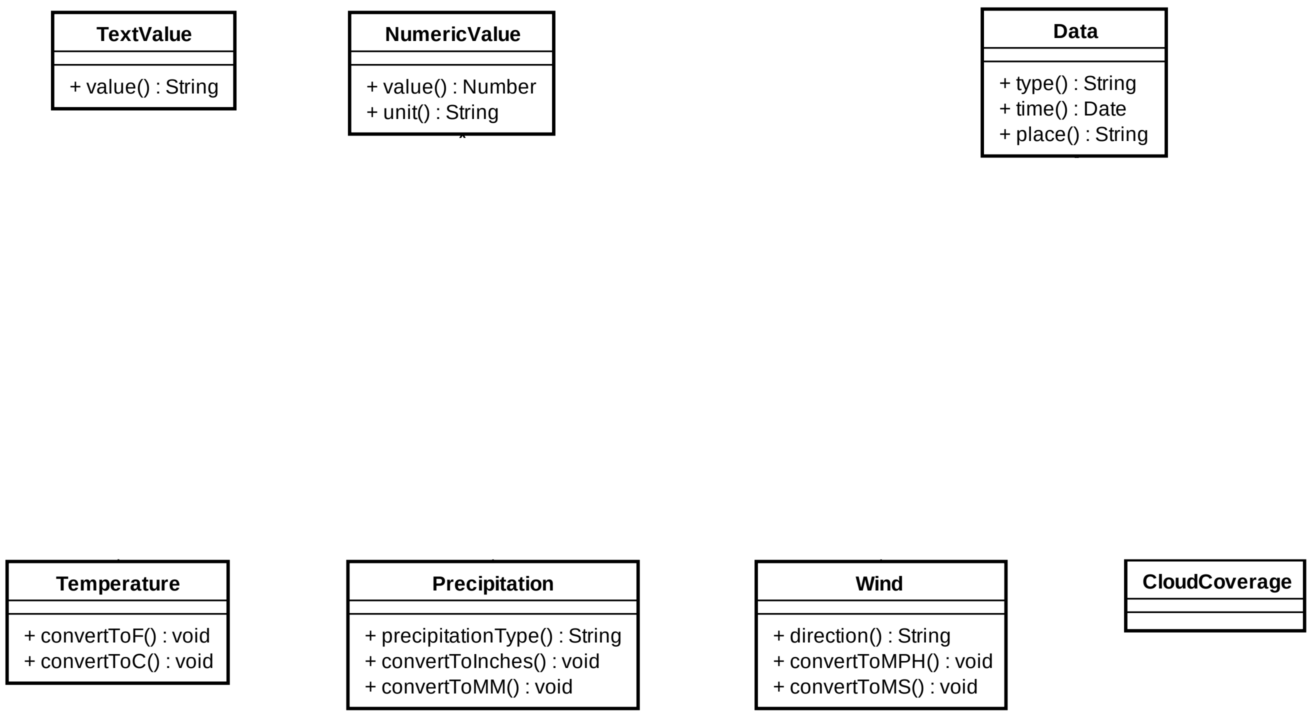
b) Create an array of circles. Use the array map() method to create an array with the radius of each circle.

c) We want to add an overloaded constructor to Circle:

```
    Circle(x: double, y: double, radius: double)
```

JavaScript doesn't support overloading like Java. How do you implement this? (Hint: use the arguments object.)

Exercise 1.3



Implement the diagram above using factory methods. Do *not* use constructors or the `class` keyword. Encapsulate everything.

Exercise 1.4 – JavaScript Recap

Exercise 1.4.1 – truthy

Which of the following are truthy?

```
a) 2 + 2 === 4
b) 2 + 2 === '4'
c) 2 + 2 == '4'
d) Number('4')
e) Number('0')
f) NaN
g) NaN != NaN
h) Infinity == Infinity
i) 1/0 == 2/0
j) 2 * null
k) 2 + null
l) 7
m) null || 7
n) '4'
o) ''
```

Exercise 1.4.2 – loops

a) Make a loop that prints (using `console.log`) the numbers from 1 to 10
b) Make a loop that adds the numbers from 1 to 10
c) Make a loop that computes 10! (factorial)

Exercise 1.4.3 – arrays

`var a = [1, 2, 3, 5, 8]` creates an array.

`a.length` is the length of the array (5)

`a[0], …, a[4]` are the elements of the array.

a) What's `a[5]`?
b) Make a loop that prints the elements of a
c) Make a loop that adds the elements of a
d) Make a function that takes an array and returns the sum of its elements
e) Add an element to a like this: `a[8] = 55`
f) What's `a[8]`?
g) What's the length of a?
h) What happens if you print a to the console?
i) What happens with your loop from (c)?

Exercise 1.4.4 – basic functions

a) Make a function, `factorial`, that takes a value n and returns n!
b) Make a function, `power`, that takes values m and n and returns m^n.

Exercise 1.4.5 – advanced functions

Make a function that takes two arguments, m and n. If n is undefined, the function should return m!, otherwise the function should return m^n.
What happens if you call the function with only one argument?