

1) Create an iffe that returns an object with fields: function setValue(), function showValue() and function reverseValue(). Calling functions either logs the value or reverse it in an object. If value was not provided yet or is empty showValue function is to return information about that. Value can be type string or number. reverseValue(): If number do $(*(-1))$, if string reverse it. **Closure pattern.**

2) Create four function definitions. One for every basic math operations and taking two input parameters. Create one more function. This function is to return calculation object. This object is to have parameters object field that holds two operation parameters inside (x and y) and a function field that points to a function with math operation (defined at the beginning). A function setOperation() that sets the field from previous sentence and a calculate() function that makes calculation and returns its value.

3) Create an array (by hand) of objects and call sum() function in context of each one of them. Sum function is to come from this object BaseObject = { x,y, sum: function () { return this.x+this.y}}

Example array: [{x:2,y:3},{x:-1,x:6},{x:0,x:8},.....]

4) Given an array of objects, for each object call operation() function in context of next object. If object is last, got back to start of the array for operation function. In loop;

```
[
  {
    x: 1,
    y: 'object one value',
    operation: () => 'object one prafix' + this.x + this.y
  },
  {
    x: 2,
    y: 'object two value',
    operation: () => 'object two prefix' + this.x + this.y
  },
  {
    x: 3,
    y: 'object three value',
    operation: () => 'object three prefix' + this.x + this.y
  },
]
```

5) Scale riddle. With 8 balls ☺ EXAM [1,2,1,1,1,1,1,1]. One of the items will be change to two. Indexes are t be chosen at random. Use comparing of weights only two times.