

2/18/2022

Assignment for w3d5 Advanced functions integration

The assignment for today is:

- Complete the review programming exercises given below
- Preread the material on prototypal inheritance
- Catch up on anything needed from the past week

In the following code the transactionsDB is publicly accessible to any code that has access to the bank object. Instead of using the object literal for bank, write a makeBank function that will encapsulate and return the bank object. Make the transactionsDB private by making it a local variable in the makeBank function instead of a property on the bank object. (Functions that construct and return complex objects like this are often call “object factories.”) Use the [mocha test file, bankTests.js](#) to test your solution.

If you are unsure of what this means to make transactionsDB a private variable review the makeCounter example. We want to access transactionsDB in an analogous manner to how we accessed counter. Even better, review the "[Counter object](#)" task.

```
const bank = {  transactionsDB: [],};
bank.transactionsDB = [
  { customerId: 1, customerTransactions: [10, 50, -40] },
  { customerId: 2, customerTransactions: [10, 10, -10] },
  { customerId: 3, customerTransactions: [5, -5, 55] }];

bank.getBalance = function (id) {
  const customer = bank.transactionsDB.find(customer => customer.customerId === id);
  let balance = 0;
  for (const trans of customer.customerTransactions) {  balance += trans;  }
  return balance;  };

/** * @returns {number} returns sum of all balances */
bank.bankBalance = function () { //IMPLEMENT THIS
```

Practice Review Exercises

The following are programming exercises for more practice on recent lessons. Use [the Mocha test file, funPracticeTests.js](#), for your map, filter, reduce functions.

1. Write your own version of map. Write a function, myMap that takes 2 arguments, an array and a function to apply to the array. It should return a new array of the same size with the function applied to each element of the input array. It should not change the input array.
2. Write your own version of filter. Write a function, myFilter that takes 2 arguments, an array and a function to apply to the array. It should return a new array with the function applied to each element of the input array. It should not change the input array. It should work like Array.filter. I.e., the input function returns true or false for each element in the original array, and the true elements are included in the returned array.
3. Write your own version of reduce, myReduce.
4. Implement the student grades problem given in the Mocha test file, gradeTests.js, in my public repository: 303Demos\mochaTests\labW3D6
5. Extra Credit (Optional) Write your own Mocha-like framework. You will need to implement your own “describe” and “it” functions, and an assert.strictEquals function. Do not worry about other assert functions. Your framework should log to the console the describe and it text messages and whether the tests fail or succeed. Try using your framework with one or two of your assignment files instead of the given Mocha test files.