1. Create an array of employee objects with name and salary properties. Create 3 functions

* First function to modify the salary of all employees by 1000 and return an array of modified employees object.
* Second function will check whether an employee’s salary is greater than 50000 or not.
* If all employees are getting salary above 50000 then ‘Best salary paying company’ is to be printed else ‘Not best salary paying company’
* Third function will check whether an employee’s salary is greater than 55000 or not and accordingly those records will get printed.

This is to be solved using the **Array Helper Functions**.

1. Find out the average salary of the employees and then print the details of those employees whose salary is greater than the average salary
2. Write a program to create a function which will accept n numbers and calculate their sum.
3. Write a program to find out the minimum number in an array using Math.min() function.
4. Write a function to which an employee object will be passed and will print only the name and department of the employee. It is to be done for all employee.
5. Complete the functions in following program

var pets={

names:['Baron','Cheif','Axel'],

owner:['Jason'],

description:function()

{

},

individual\_\_pet\_object: function()

{

}

* “description” function should return the description in following format

Jason knows an awesome dog named Baron

Jason knows an awesome dog named Cheif

Jason knows an awesome dog named Axel

* “individual\_pet\_object” function should compose and return objects in following format

{name: "Baron", owner: "Jason"}

{name: "Cheif", owner: "Jason"}

{name: "Axel", owner: "Jason"}

1. Create an Account class with following

* Constructor with two properties balance and account number
* Functions
  + ‘withdraw’ with one parameter which will deduct from balance
  + ‘deposit’ with one parameter which will add to balance
  + ‘getBalance’ to check the balance
  + Getter and setter for account number
  + ‘toString’ to print the account details.

1. Using the Account class as a base class, write two derived classes called SavingsAccount and CurrentAccount. A SavingsAccount object, in addition to the attributes of an Account object, should have an interest variable and a method which adds interest to the account. A CurrentAccount object, in addition to the attributes of an Account object, should have an overdraft limit variable. Ensure that you have overridden methods of the Account class as necessary in both derived classes.
2. Write a program to create a SET with the existing set of values in an array
3. Write a program to iterate through all elements in a SET.
4. Write a program to get the number of elements in a SET.
5. Write a program to empty a SET
6. Write a program to test a SET is empty or not.
7. Write a program to clone a SET to another SET.
8. Write a program to convert a SET to an array.
9. Write a program to check whether all elements in one SET is present in another or not.
10. Write a program to create an empty Map and then after add three employees object with 1,2,3 as their key.
11. Write a program to count the number of key-value (size) mappings in a map.
12. Write a program to copy all of the mappings from the specified map to another map.
13. Write a program to remove all of the mappings from a map.
14. Write a program to check whether a map contains key-value mappings (empty) or not.
15. Write a program to test if a map contains a mapping for the specified key.
16. Write a Java program to get the value of a specified key in a map.
17. Write a program to get a list of the keys contained in the map.
18. Write a program to get a list of the values contained in the map.