Computer Language

Assignment

- Complete the following MyArray class in TypeScript that can handle string and number only to get the expected results

Barebone class definition

```
class MyArray {
  add(value) {
    this.collection.push(value)
  }

  remove(value) {}

  getValues() {
    return this.collection
  }
}
```

Expected results

```
const stringAry = new MyArray()
stringAry.add('aaa')
stringAry.add('bbb')
stringAry.add('ccc')
stringAry.remove('bbb')
console.log(stringAry.getValues()) //=> [ 'aaa', 'ccc' ]

const numberAry = new MyArray()
numberAry.add(1)
numberAry.add(2)
numberAry.add(3)
numberAry.remove(2)
console.log(numberAry.getValues()) //=> [ 1, 3 ]

const booleanAry = new MyArray()
numberAry.add(true)
numberAry.add(false) //=> You shouldn't be able to do these.
```

- Create an arrow function that takes an object with default values. Here're the interface and expected results.

```
interface Params {
  firstParam: string
  secondParam: string
}

console.log(myFunc()) //=> { firstParam: 'defaultFirst', secondParam:
'defaultSecond' } // The default values are returned when args not given.

console.log(myFunc({ firstParam: 'first', secondParam: 'second' })) //=> {
firstParam: 'first', secondParam: 'second' } // The given values are
returned when args given.

console.log(myFunc({ firstParam: 'first' })) //=> { firstParam: 'first',
secondParam: 'defaultSecond' }

console.log(myFunc({ secondParam: 'second' })) //=> { firstParam:
'defaultFirst', secondParam: 'second' }
```

- Complete the following function called validator, that takes an optional function argument. It caches the response and return it when there's.

Given interface and function definitions

```
interface CustomResponse {
  valid: boolean
  error: string
}

const errorResponse = (): CustomResponse => {
  return { valid: false, error: 'error occurred' }
}

// Complete this fuction to work described in the following expected
results. const validator = ( validatorFn?: () => CustomResponse ):
CustomResponse => { }
```

Expected results

```
console.log(validator(() => errorResponse()).valid) //=> false

console.log(validator().error) //=> 'error occurred'
```

- write typescript code with following requirements (design proper classes). The requirements have suggestions for fields and methods, you are free to add more as per your thinking
    - Bank Account

        - fields: id, firstName, lastName, address, phone, email, type (saving/current)
        - methods: createAccount, updateAccount, deleteAccount
    - Transaction
        - fields: date, type, amount, customerId
        - methods: depositFunds, withdrawFunds
- Write a program that asks the user how many days are in a particular month, and what day of the week the month begins on (0 for Monday, 1 for Tuesday, etc), and then prints a calendar for that month. For example, here is the output for a 30-day month that begins on day 4 (Thursday)
- Write a program that contains a function that has one parameter, n, representing an integer greater than 0. The function should return n! (n factorial). Then write a main function that calls this function with the values 1 through 20, one at a time, printing the returned results. This is what your output should look like:
  126
  24
  120
  720
  5040
  40320
  362880
  36288002
- Use inheritance
    - Write a class Course with name,fees. Provide following functionalities
        - initializer
        - Accept data
        - Print Data .
    - Write a Class Computer with subjectList.
      Provide following functionalities
        - initializer
        - Accept data
        - Print Data .
    - Write a Class Electonics with subjectList.
      Provide following functionalities
        - initializer
        - Accept data
        - Print Data
- A pangram is a sentence that contains all the letters of the English alphabet at least once, for example: The quick brown fox jumps over the lazy dog. Your task here is to write a function to

check a sentence to see if it is a pangram or not.

- Write a function translate() that will translate a text into "rövarspråket" (Swedish for "robber's language"). That is, double every consonant and place an occurrence of "o" in between. For example, translate("this is fun") should return the string "tothohisos isos fofunon".
- Create a class 'Student' with rollno, studentName, course ,dictionary of marks(subjectName ->marks [5]).
  Provide following functionalities
  - initializer
  - override **str** method
  - accept student data
  - Print student data for given id.
  - Print Student who has failed in any subject.
  - Write menu driven console program to test above functionalities.(accept records of 5 students and store those in list)
- Write a program to Interchange First and Last Element of a List
- Write a python program to print sum of tuple elements
- Replace single element ‘b’ in given list ['a', 'b', 'c', 'd', 'e'] with [1, 2, 3]
- write a program to find index of element ‘e’ in given vowels list ['a', 'e', 'i', 'o', 'i', 'u']
- Accept the full name from user(Name Middlename Surname) in lowercase and Print it in title case. NOTE:(Using in-built function)