

```

'use strict';
const dayjs = require('dayjs');
const isSameOrAfter = require('dayjs/plugin/
isSameOrAfter');
dayjs.extend(isSameOrAfter);

/**
 * Constructor function for new Exam objects
 * @param {string} code course code (e.g., '01ABC')
 * @param {string} name course name
 * @param {number} cfu number of credits (e.g. 6)
 * @param {number} score score attained at the exam
 * @param {string} date date of the exam, in a format
parseable by dayjs()
 */
function Exam(code, name, cfu, score, date) {
    this.code = code;
    this.name = name;
    this.cfu = cfu;
    this.score = score;
    this.date = dayjs(date);
}

/**
 * Alternative form for a function to create an exam
 * @param {string} code course code (e.g., '01ABC')
 * @param {string} name course name
 * @param {number} cfu number of credits (e.g. 6)
 * @param {number} score score attained at the exam
 * @param {string} date date of the exam, in a format
parseable by dayjs()
 * @return {Exam} a newly created Exam object
 */
function createExam(code, name, cfu, score, date) {
    return {
        code: code,
        name: name,

```

```

        cfu: cfu,
        score: score,
        date: dayjs(date),
    };
}

/**
 * Create a new empty transcript object (a list of Exam
objects)
 */
function ExamList() {
    this.examList = [];

    /**
     * Add a new exam to the list
     * @param {Exam} exam the exam to be added
     */
    this.add = (exam) => {
        this.examList.push(exam);
    };

    this.find = (code) => {
        const result =
this.examList.filter((exam)=>exam.code===code);
        return result.length ? result[0] : undefined;

        // if(result.length)
        //     return result[0];
        // else
        //     return undefined;

        // for(let exam of this.examList) {
        //     if(exam.code === code) {
        //         return exam ;
        //     }
        // }
        // return undefined;
    };
};

```

```

    // this.afterDate = (startDate) => {
    //     const result = new ExamList() ;
    //     const startDay = dayjs(startDate) ;
    //     for(let exam of this.examList) {
    //
    if( exam.date.isSameOrAfter(startDay) ) //compare
startDay with exam.date is ok
        //         result.add(exam);
    //     }
    //     return result ;
    // }

```

```

this.afterDate = (startDate) => {
    const result = new ExamList();
    const startDay = dayjs(startDate);
    result.examList = this.examList.filter(
        (exam) =>
exam.date.isSameOrAfter(startDay));
    return result;
};

```

```

this.print = () => {
    this.examList.forEach( (exam, i) => {
        console.log(`${i+1}) ${exam.name} - $
{exam.score}`);
    });
};

```

```

// this.average = () => {
//     // const scores = this.examList.map(exam =>
exam.score);
//     // console.log(scores)
//     // const sum = scores.reduce((partial,
current)=>partial+current, 0)
//     // console.log(sum)

```

```

        //      return this.examList.map(exam =>
exam.score).reduce((partial, current)=>partial+current,
0)
        //      / this.examList.length
        // }
        this.average = () => this.examList.map((exam) =>
exam.score)
        .reduce((partial, current)=>partial+current,
0) / this.examList.length;
    }

```

```

const myexam = new Exam('12abc', 'whatever course', 8,
27, '2022-03-08');
// console.log(myexam) ;

```

```

const myotherexam = createExam('89xyz', 'useless
course', 12, 29, '2022-04-08');
// console.log(myotherexam) ;

```

```

// console.log(dayjs('2022-03-08'))

```

```

const allMyExams = new ExamList();
allMyExams.add(myexam);
allMyExams.add(myotherexam);
allMyExams.add(new Exam('66zzz', 'title', 6, 24,
'2021-09-01'));

```

```

allMyExams.print();

```

```

// const examsIn2022 =
allMyExams.afterDate('2022-01-01');

```

```

console.log(allMyExams.average());

```

```

debugger;

```