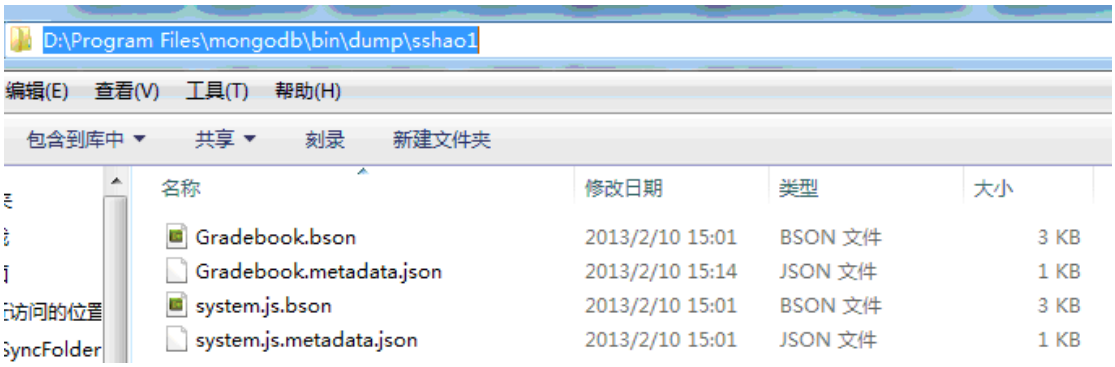
CSE446/598 Assignment1

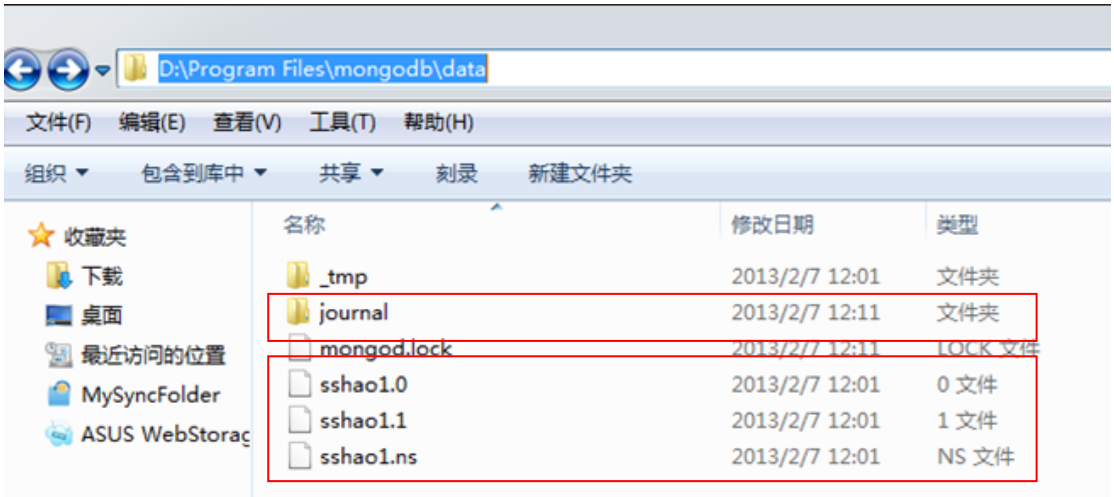
Readme

Shihuan Shao

1. Download the zipped package “Shihuan Shao final submission”. In the folder of mongodb (suppose the path is D:\Program Files\mongodb), enter the bin folder. Copy the dump folder which contains a folder called sshao1 into the bin folder. In sshao1, there are four bson/json files, which are necessary to restore the database. You also need to create a folder called data in the folder mongodb.

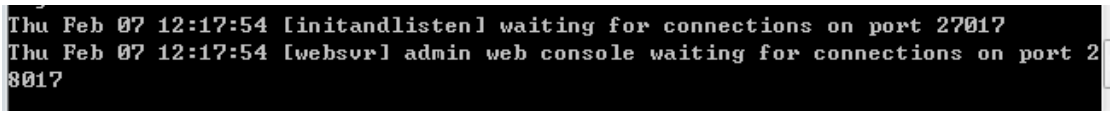


Another way to restore the db in case the mongorestore does not work: Create a folder called data in the folder mongodb and copy the files in the alternative folder into the data folder.



2. Run cmd and enter the bin folder of mongodb
D:\Program Files\mongodb\bin.
Then type mongod –dbpath “D:\Program Files\mongodb\data”

The following figure means you started the mongodb successfully.



3. Keep the mongodb running. Now run another cmd, and enter the bin folder of mongodb.
Restore the database:

```
mongorestore --db sshao1 "D:\Program Files\mongodb\bin\dump\sshao1"
```

```
D:\Program Files\mongodb\bin>mongorestore --db sshao1 "D:\Program Files\mongodb\
bin\dump\sshao1"
connected to: 127.0.0.1
Sun Feb 10 15:26:35 D:/Program Files/mongodb/bin/dump/sshao1/Gradebook.bson
Sun Feb 10 15:26:35     going into namespace [sshao1.Gradebook]
Sun Feb 10 15:26:35 warning: Restoring to sshao1.Gradebook without dropping. Res
tored data will be inserted without raising errors; check your server log
11 objects found
Sun Feb 10 15:26:35 D:/Program Files/mongodb/bin/dump/sshao1/system.js.bson
Sun Feb 10 15:26:35     going into namespace [sshao1.system.js]
Sun Feb 10 15:26:35 warning: Restoring to sshao1.system.js without dropping. Res
tored data will be inserted without raising errors; check your server log
1 objects found
Sun Feb 10 15:26:35     Creating index: { key: { _id: 1 }, ns: "sshao1.system.js
", name: "_id_" }
```

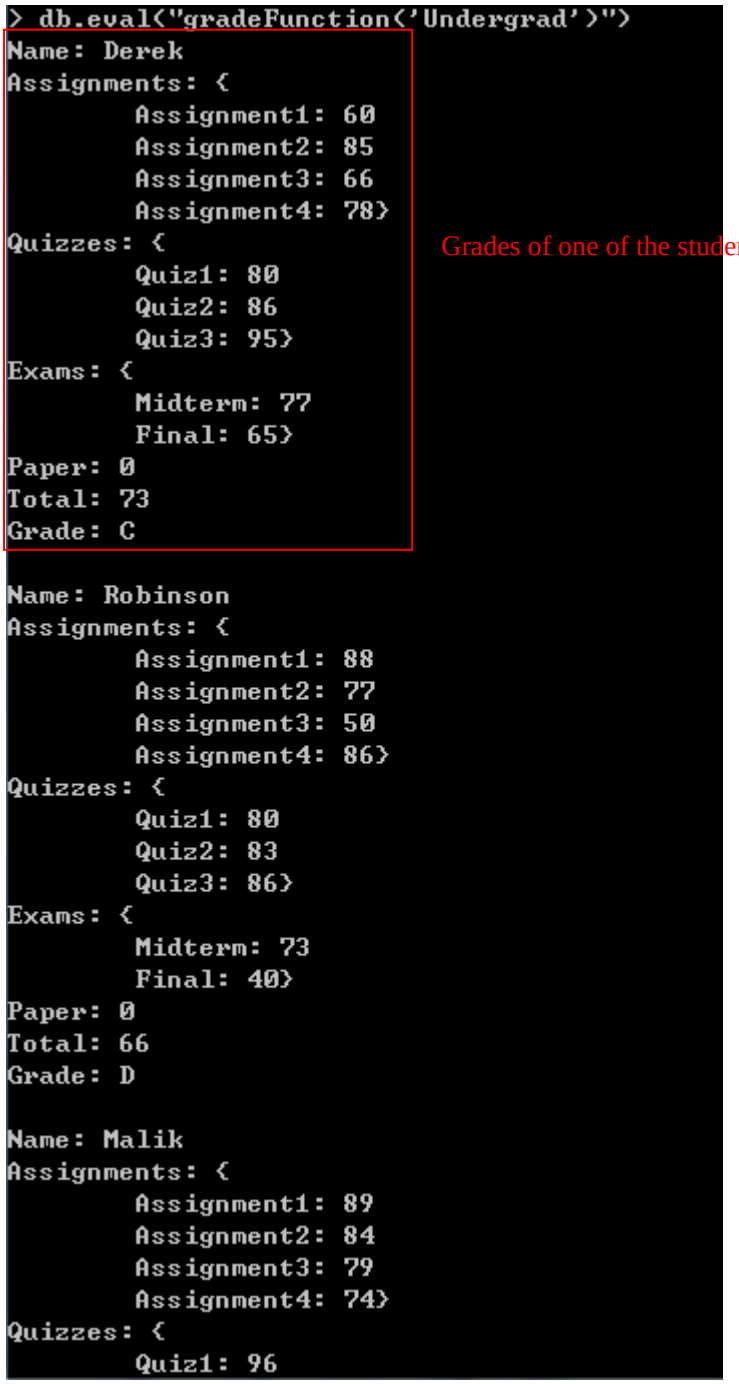
Now the database is restored. Then type `mongo` to start the mongo shell, or you can directly click the mongo.exe. You will see:

```
D:\Program Files\mongodb\bin>mongo
MongoDB shell version: 2.2.2
connecting to: test
>
```

Then switch to database sshao1 using instruction `use sshao1`

```
D:\Program Files\mongodb\bin>mongo
MongoDB shell version: 2.2.2
connecting to: test
> use sshao1
switched to db sshao1
>
```

4. Now you can do whatever you want.
(1) If you want to list all undergrad students and their grades, use the instruction
`db.eval("gradeFunction('Undergrad')")`
Undergrad (Note: "U" is uppercase) is student category parameter. By using this instruction, the grading script will be automatically called to calculate the student's total score and letter grade.



Grades of one of the students

(2) If you want to list all grad students and their grades, use the instruction
`db.eval("gradeFunction('Grad')")` Note: “G” in “**G**rad” is also uppercase.

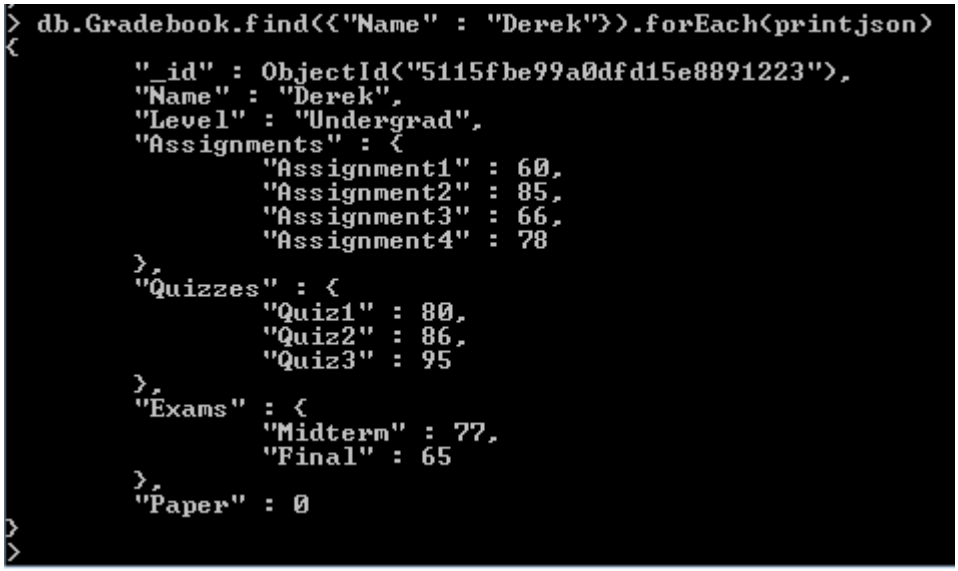
```
> db.eval("gradeFunction('Grad')")
Name: Shihuan
Assignments: {
        Assignment1: 94
        Assignment2: 92
        Assignment3: 91
        Assignment4: 90}
Quizzes: {
        Quiz1: 95
        Quiz2: 98
        Quiz3: 95}
Exams: {
        Midterm: 94
        Final: 96}
Paper: 90
Total: 93
Grade: A-

Name: Michael
Assignments: {
        Assignment1: 97
        Assignment2: 98
        Assignment3: 98
        Assignment4: 84}
Quizzes: {
        Quiz1: 90
        Quiz2: 95
        Quiz3: 96}
Exams: {
        Midterm: 89
        Final: 90}
Paper: 85
Total: 90
Grade: A-

Name: Keith
Assignments: {
        Assignment1: 99
        Assignment2: 87
        Assignment3: 95
        Assignment4: 93}
Quizzes: {
```

(3) To query students called “Derek”, type

```
db.Gradebook.find({"Name" : "Derek"}).forEach(printjson)
```



(4) To add a new document for student Chris and his grades, you need to first create a document:

```
Chris = {Name : "Chris", Level : "Undergrad", Assignments : {Assignment1 : 60, Assignment2 : 85, Assignment3 : 66, Assignment4 : 78}, Quizzes : {Quiz1 : 80, Quiz2 : 86, Quiz3 : 95 }, Exams : {Midterm : 77, Final : 65}, Paper : 0};
```

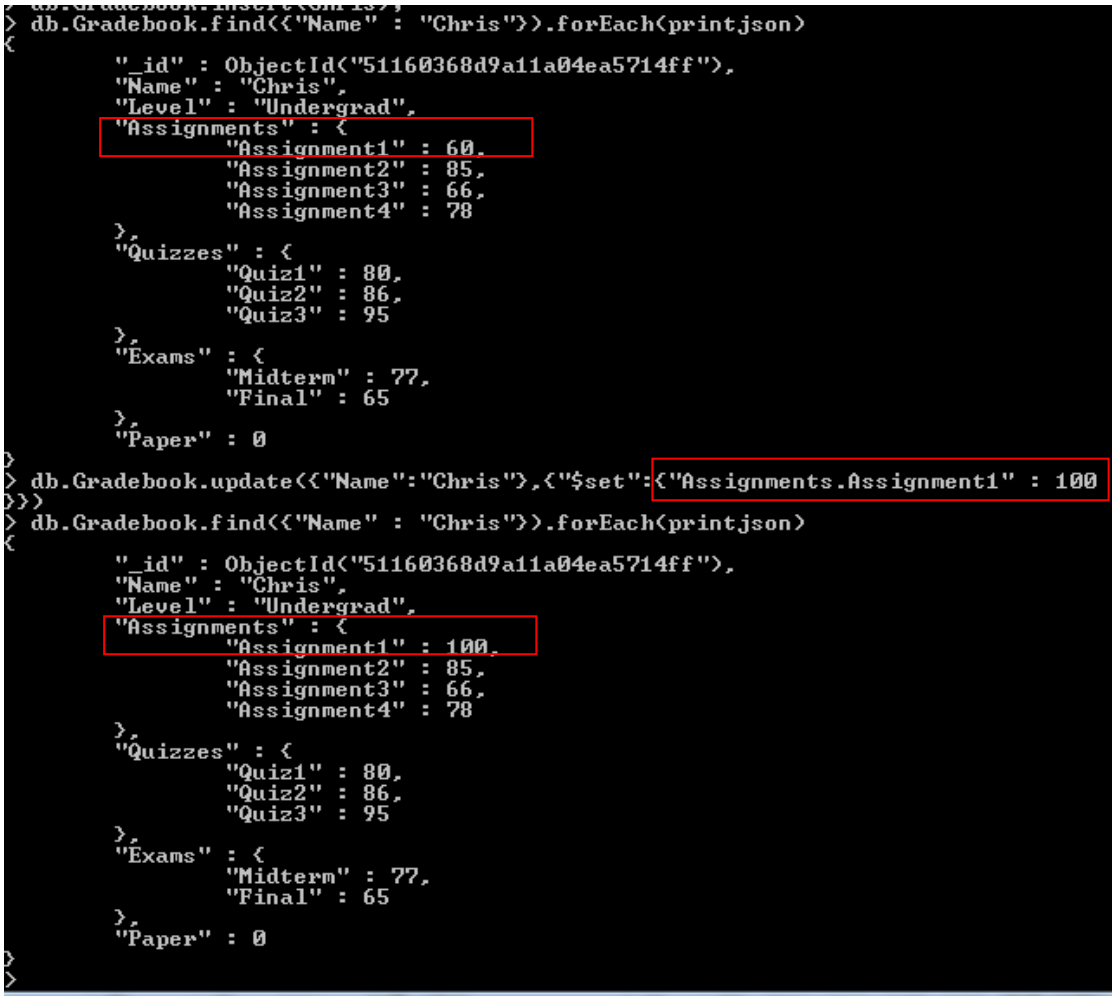
Note: The instruction above should be typed in one line.

Then `db.Gradebook.insert(Chris);`

```
> Chris = {Name : "Chris", Level : "Undergrad", Assignments : {Assignment1 : 60,
 Assignment2 : 85, Assignment3 : 66,   Assignment4 : 78}, Quizzes : {Quiz1 : 80,
 Quiz2 : 86, Quiz3 : 95 }, Exams : {Midterm : 77, Final : 65}, Paper : 0};
{
        "Name" : "Chris",
        "Level" : "Undergrad",
        "Assignments" : {
                "Assignment1" : 60,
                "Assignment2" : 85,
                "Assignment3" : 66,
                "Assignment4" : 78
        },
        "Quizzes" : {
                "Quiz1" : 80,
                "Quiz2" : 86,
                "Quiz3" : 95
        },
        "Exams" : {
                "Midterm" : 77,
                "Final" : 65
        },
        "Paper" : 0
}
> db.Gradebook.insert(Chris);
> db.Gradebook.find({"Name" : "Chris"}).forEach(printjson)
{
        "_id" : ObjectId("51160368d9a11a04ea5714ff"),
        "Name" : "Chris",
        "Level" : "Undergrad",
        "Assignments" : {
                "Assignment1" : 60,
                "Assignment2" : 85,
                "Assignment3" : 66,
                "Assignment4" : 78
        },
        "Quizzes" : {
                "Quiz1" : 80,
                "Quiz2" : 86,
                "Quiz3" : 95
        },
        "Exams" : {
                "Midterm" : 77,
                "Final" : 65
        },
        "Paper" : 0
}
>
```

(5) To update Chris's grade of Assignment1:

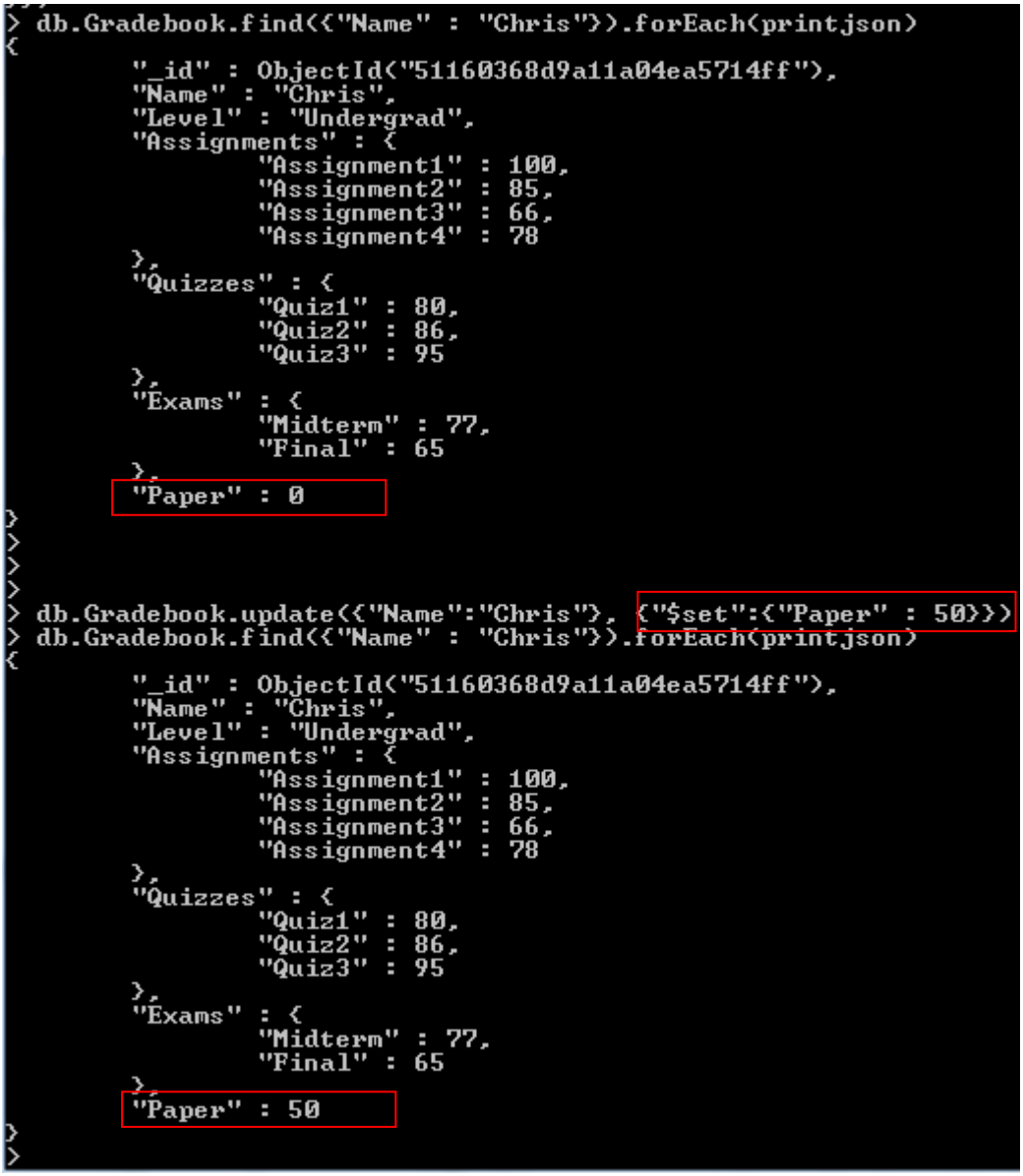
```
db.Gradebook.update({"Name":"Chris"},{"$set":{"Assignments.Assignment1" : 100 }})
```



Because of the structual similiarity, the update of quiz scores and exam scores are the same.

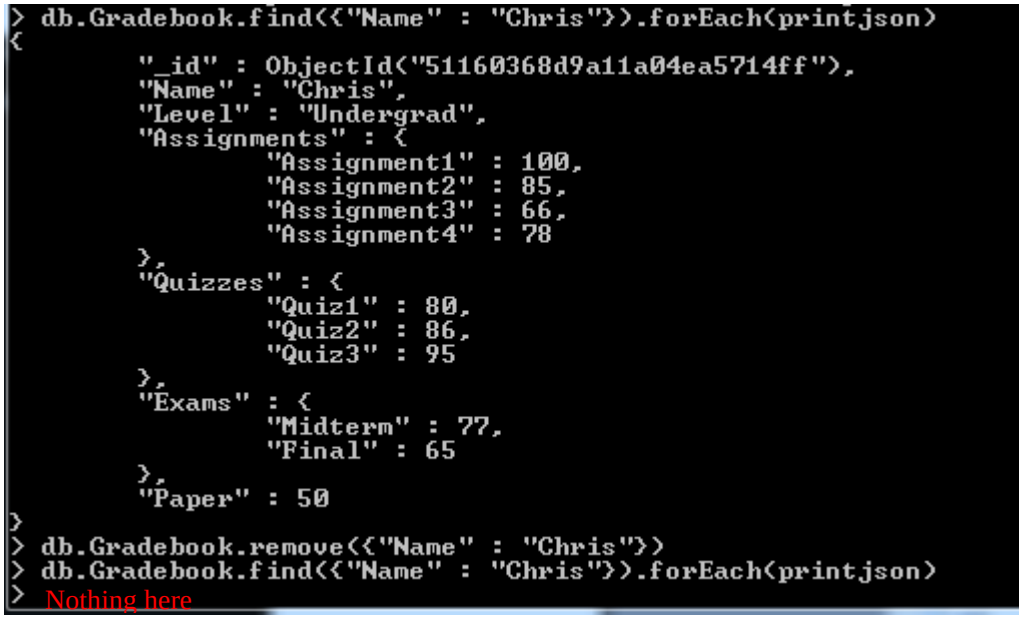
(6) To update Chris's grade of paper:

```
db.Gradebook.update({"Name":"Chris"}, {"$set":{"Paper" : 50}})
```

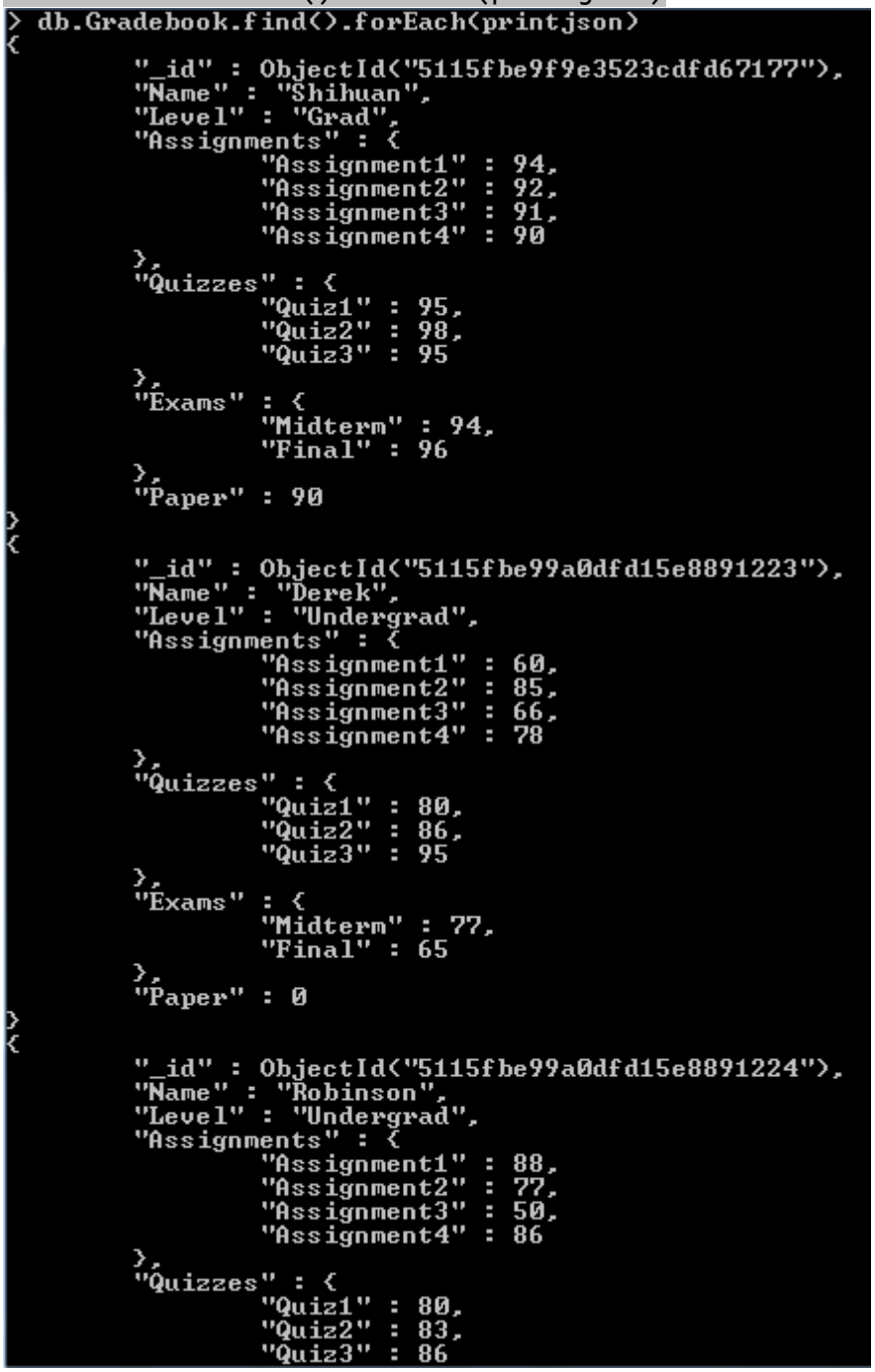


(7) To remove the document of Chris:

```
db.Gradebook.remove({"Name" : "Chris"})
```



(8) To list all the documents in the collection sshao1,
db.Gradebook.find().forEach(printjson)



-End-