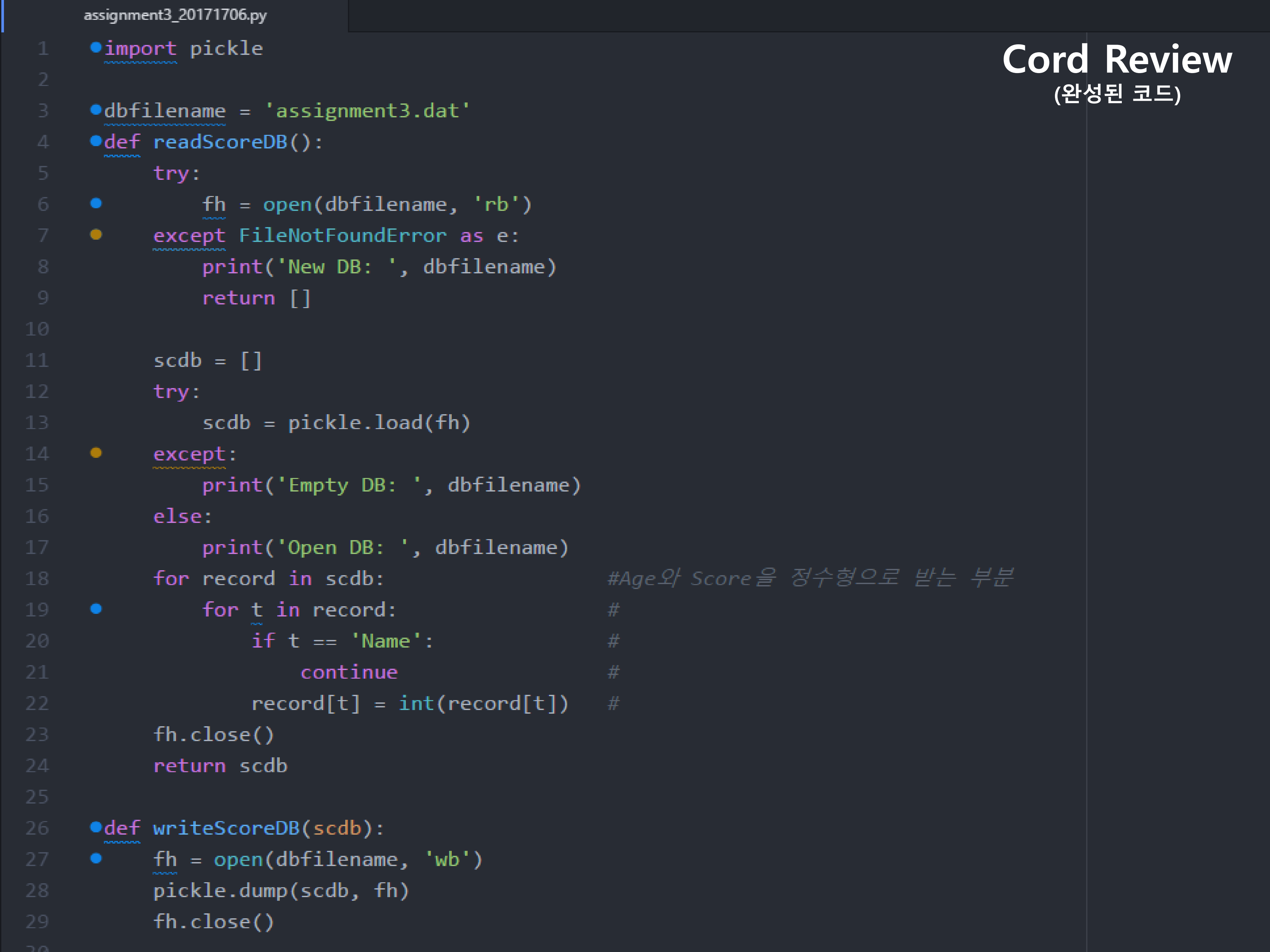
소프트웨어 프로젝트

1조

정진혁, 조가성, 조경상, 조상연

Cord Review

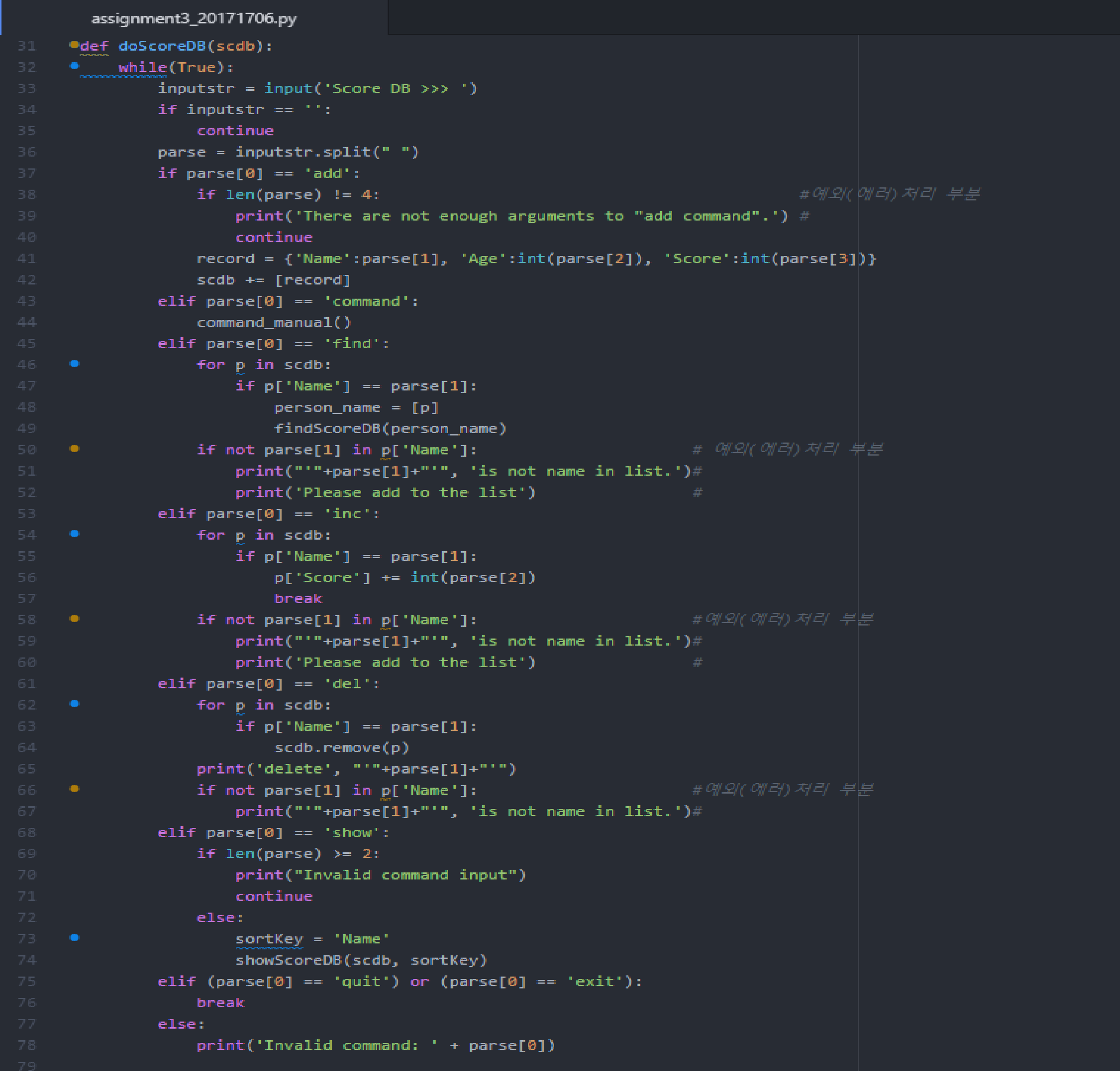
(완성된 코드)



```python
import pickle

dbfilename = 'assignment3.dat'
def readScoreDB():
    try:
        fh = open(dbfilename, 'rb')
    except FileNotFoundError as e:
        print('New DB: ', dbfilename)
        return []

    scdb = []
    try:
        scdb = pickle.load(fh)
    except:
        print('Empty DB: ', dbfilename)
    else:
        print('Open DB: ', dbfilename)
    for record in scdb:                      #Age와 Score을 정수형으로 받는 부분
        for t in record:                     #
            if t == 'Name':                  #
                continue                     #
            record[t] = int(record[t])       #
    fh.close()
    return scdb

def writeScoreDB(scdb):
    fh = open(dbfilename, 'wb')
    pickle.dump(scdb, fh)
    fh.close()
```

Cord Review

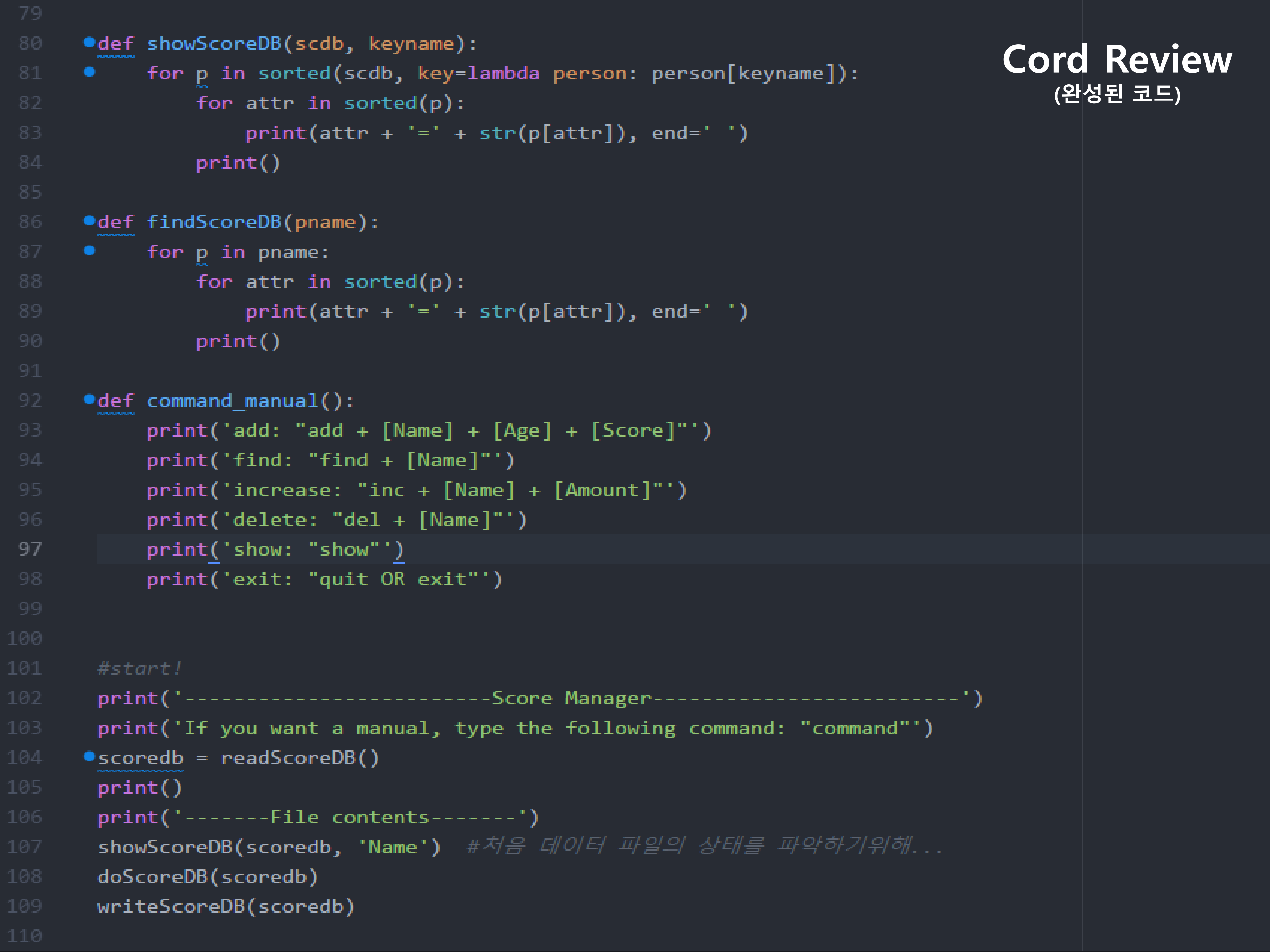
(완성된 코드)



assignment3_20171706.py

```python
def doScoreDB(scdb):
    while(True):
        inputstr = input('Score DB >>> ')
        if inputstr == '':
            continue
        parse = inputstr.split(" ")
        if parse[0] == 'add':
            if len(parse) != 4:                                           #예외(에러) 처리 부분
                print('There are not enough arguments to "add command".') #
                continue
            record = {'Name':parse[1], 'Age':int(parse[2]), 'Score':int(parse[3])}
            scdb += [record]
        elif parse[0] == 'command':
            command_manual()
        elif parse[0] == 'find':
            for p in scdb:
                if p['Name'] == parse[1]:
                    person_name = [p]
                    findScoreDB(person_name)
            if not parse[1] in p['Name']:                      # 예외(에러) 처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
                print('Please add to the list')                #
        elif parse[0] == 'inc':
            for p in scdb:
                if p['Name'] == parse[1]:
                    p['Score'] += int(parse[2])
                    break
            if not parse[1] in p['Name']:                      #예외(에러) 처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
                print('Please add to the list')                #
        elif parse[0] == 'del':
            for p in scdb:
                if p['Name'] == parse[1]:
                    scdb.remove(p)
            print('delete', "'"+parse[1]+"'")
            if not parse[1] in p['Name']:                      #예외(에러) 처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
        elif parse[0] == 'show':
            if len(parse) >= 2:
                print("Invalid command input")
                continue
            else:
                sortKey = 'Name'
                showScoreDB(scdb, sortKey)
        elif (parse[0] == 'quit') or (parse[0] == 'exit'):
            break
        else:
            print('Invalid command: ' + parse[0])
```

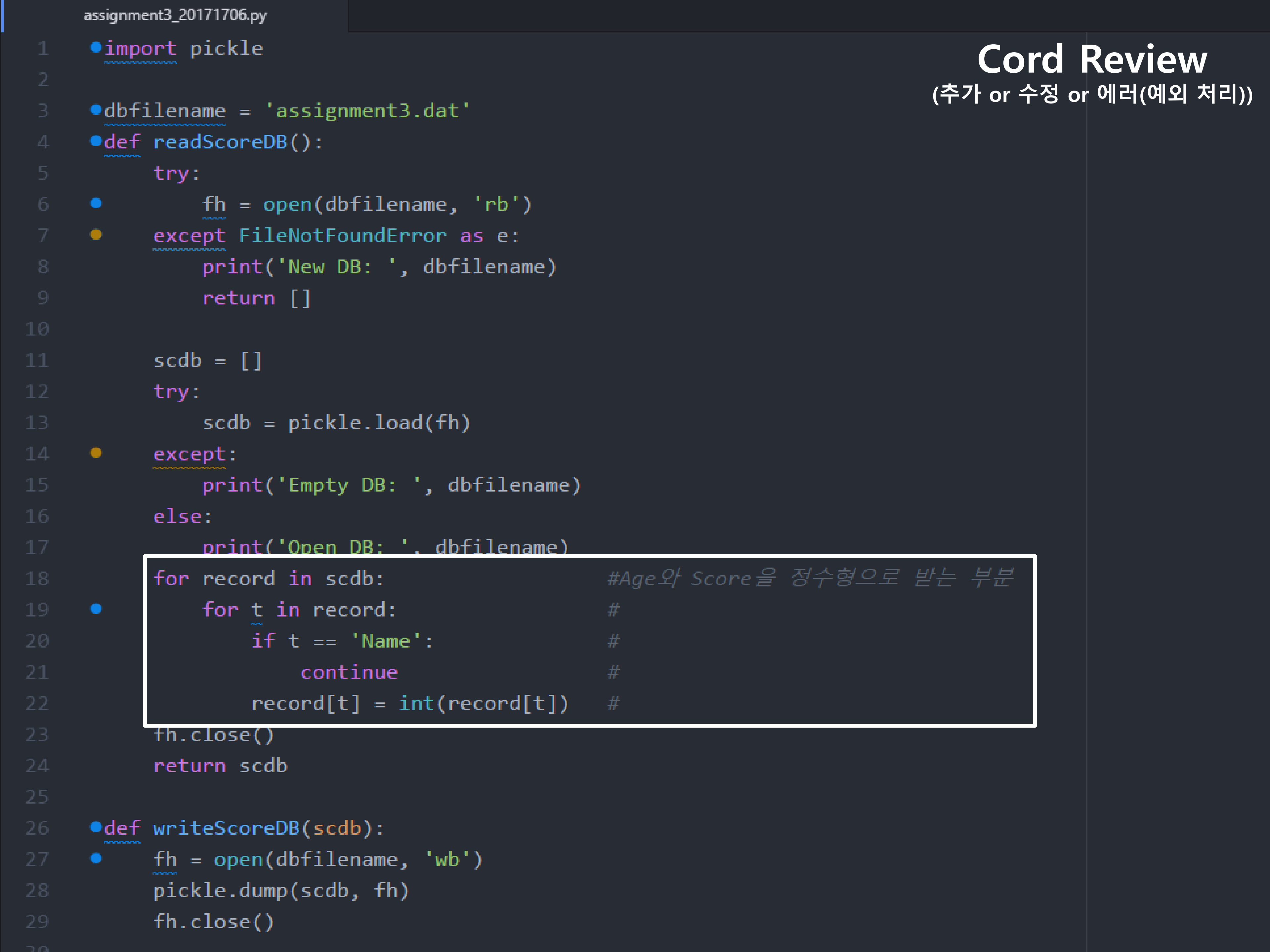
Cord Review
(완성된 코드)



```python

def showScoreDB(scdb, keyname):
    for p in sorted(scdb, key=lambda person: person[keyname]):
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()

def findScoreDB(pname):
    for p in pname:
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()

def command_manual():
    print('add: "add + [Name] + [Age] + [Score]"')
    print('find: "find + [Name]"')
    print('increase: "inc + [Name] + [Amount]"')
    print('delete: "del + [Name]"')
    print('show: "show"')
    print('exit: "quit OR exit"')


#start!
print('-------------------------Score Manager-------------------------')
print('If you want a manual, type the following command: "command"')
scoredb = readScoreDB()
print()
print('-------File contents-------')
showScoreDB(scoredb, 'Name')  #처음 데이터 파일의 상태를 파악하기위해...
doScoreDB(scoredb)
writeScoreDB(scoredb)

```

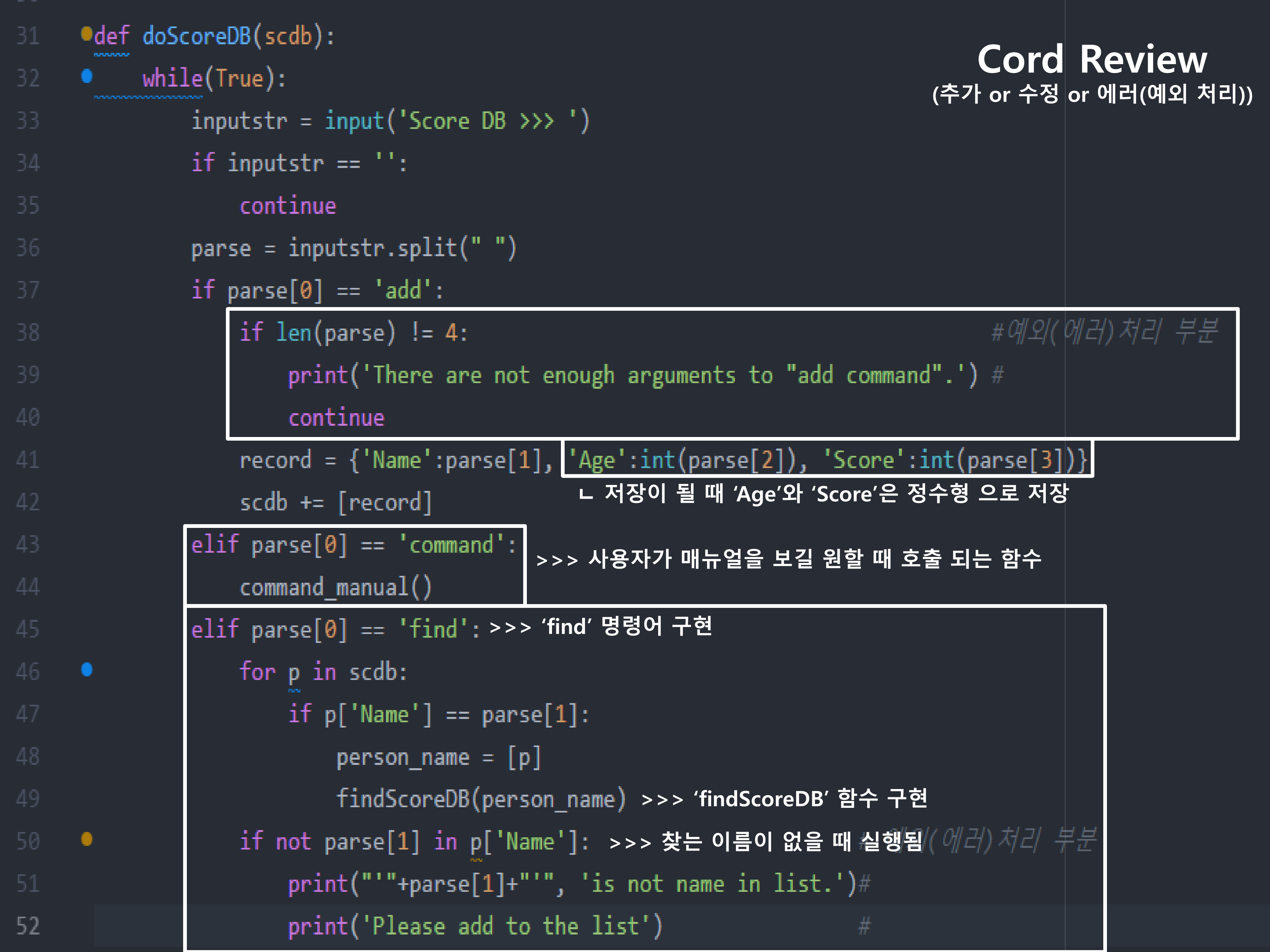
Cord Review

(추가 or 수정 or 에러(예외 처리))



assignment3_20171706.py

```python
import pickle

dbfilename = 'assignment3.dat'
def readScoreDB():
    try:
        fh = open(dbfilename, 'rb')
    except FileNotFoundError as e:
        print('New DB: ', dbfilename)
        return []

    scdb = []
    try:
        scdb = pickle.load(fh)
    except:
        print('Empty DB: ', dbfilename)
    else:
        print('Open DB: ', dbfilename)
    for record in scdb:                #Age와 Score을 정수형으로 받는 부분
        for t in record:               #
            if t == 'Name':            #
                continue               #
            record[t] = int(record[t]) #
    fh.close()
    return scdb

def writeScoreDB(scdb):
    fh = open(dbfilename, 'wb')
    pickle.dump(scdb, fh)
    fh.close()
```



Cord Review

(추가 or 수정 or 에러(예외 처리))

```python
def doScoreDB(scdb):
    while(True):
        inputstr = input('Score DB >>> ')
        if inputstr == '':
            continue
        parse = inputstr.split(" ")
        if parse[0] == 'add':
            if len(parse) != 4:                                          #예외(에러)처리 부분
                print('There are not enough arguments to "add command".') #
                continue
            record = {'Name':parse[1], 'Age':int(parse[2]), 'Score':int(parse[3])}
```

ㄴ 저장이 될 때 'Age'와 'Score'은 정수형 으로 저장

```python
            scdb += [record]
        elif parse[0] == 'command':
            command_manual()
```

>>> 사용자가 매뉴얼을 보길 원할 때 호출 되는 함수

```python
        elif parse[0] == 'find':
```

>>> 'find' 명령어 구현

```python
            for p in scdb:
                if p['Name'] == parse[1]:
                    person_name = [p]
                    findScoreDB(person_name)
```

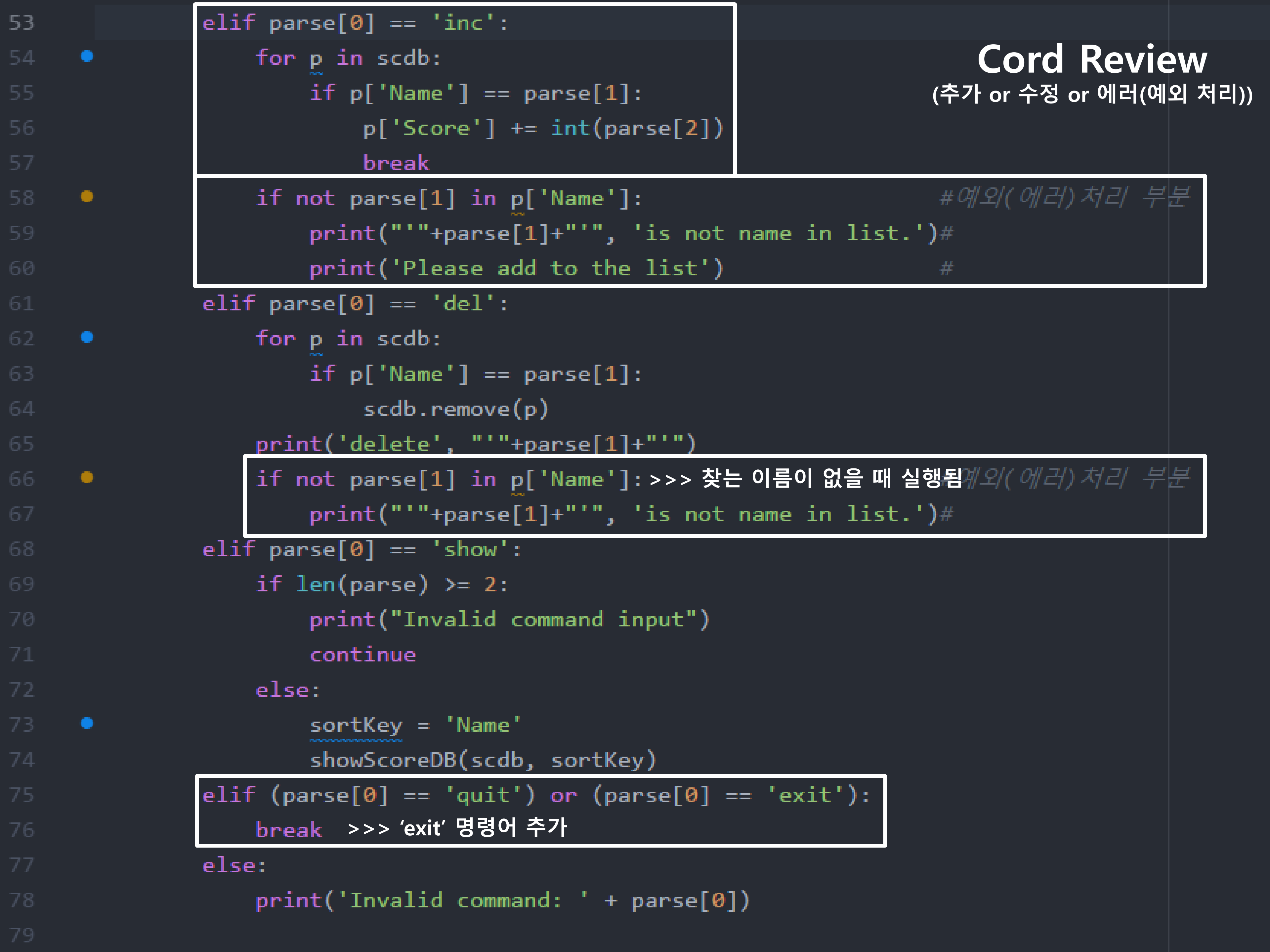
>>> 'findScoreDB' 함수 구현

```python
            if not parse[1] in p['Name']:             #예외(에러)처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
                print('Please add to the list')                #
```

>>> 찾는 이름이 없을 때 실행됨

Cord Review

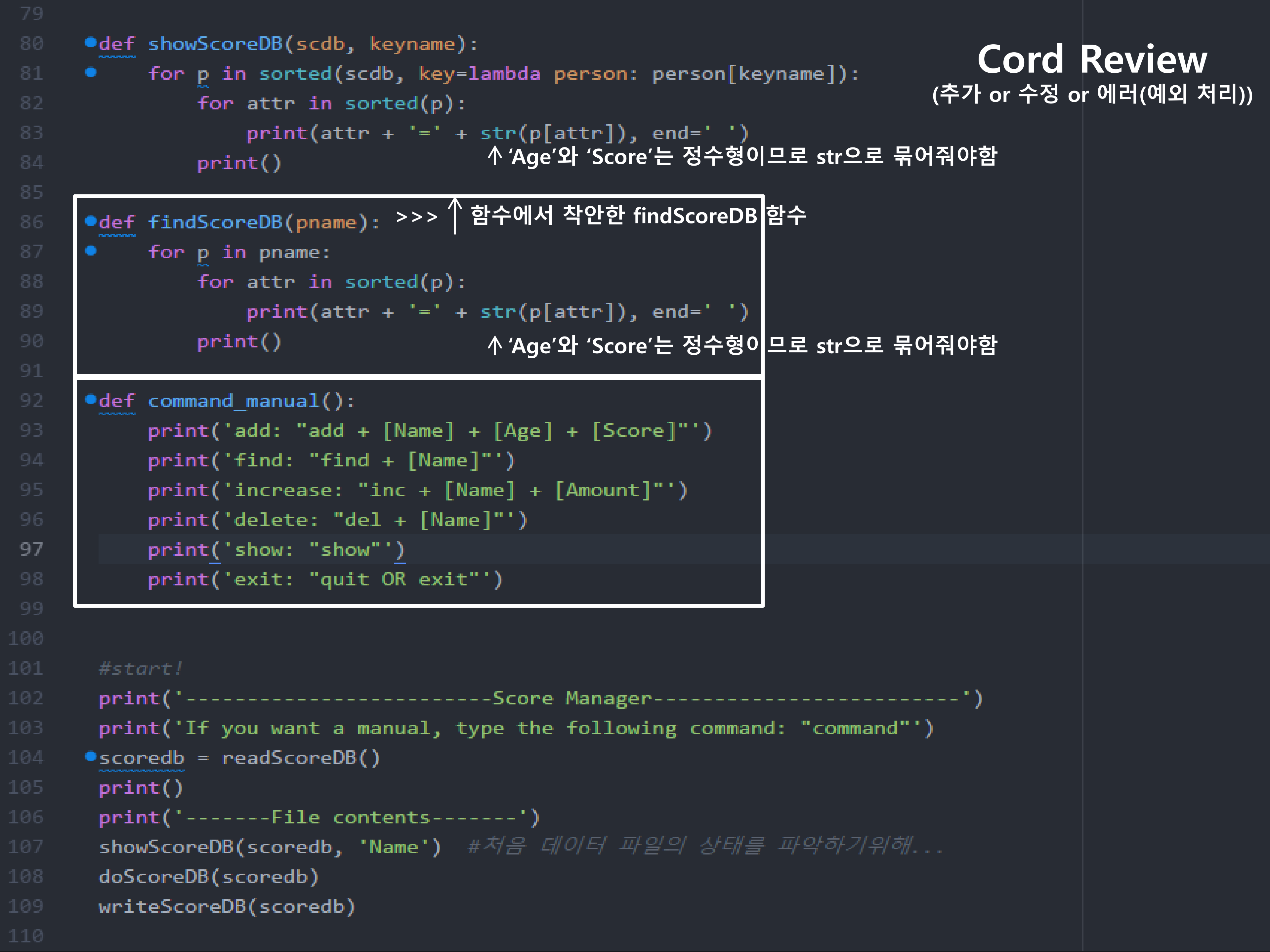
(추가 or 수정 or 에러(예외 처리))



```
        elif parse[0] == 'inc':
            for p in scdb:
                if p['Name'] == parse[1]:
                    p['Score'] += int(parse[2])
                    break
            if not parse[1] in p['Name']:                              #예외(에러)처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
                print('Please add to the list')                        #
        elif parse[0] == 'del':
            for p in scdb:
                if p['Name'] == parse[1]:
                    scdb.remove(p)
            print('delete', "'"+parse[1]+"'")
            if not parse[1] in p['Name']: >>> 찾는 이름이 없을 때 실행됨예외(에러)처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
        elif parse[0] == 'show':
            if len(parse) >= 2:
                print("Invalid command input")
                continue
            else:
                sortKey = 'Name'
                showScoreDB(scdb, sortKey)
        elif (parse[0] == 'quit') or (parse[0] == 'exit'):
            break   >>> 'exit' 명령어 추가
        else:
            print('Invalid command: ' + parse[0])
```

Cord Review

(추가 or 수정 or 에러(예외 처리))



```
def showScoreDB(scdb, keyname):
    for p in sorted(scdb, key=lambda person: person[keyname]):
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()
```

↑ 'Age'와 'Score'는 정수형이므로 str으로 묶어줘야함

```
def findScoreDB(pname):
```

>>> ↑ 함수에서 착안한 findScoreDB 함수

```
    for p in pname:
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()
```

↑ 'Age'와 'Score'는 정수형이므로 str으로 묶어줘야함

```
def command_manual():
    print('add: "add + [Name] + [Age] + [Score]"')
    print('find: "find + [Name]"')
    print('increase: "inc + [Name] + [Amount]"')
    print('delete: "del + [Name]"')
    print('show: "show"')
    print('exit: "quit OR exit"')


#start!
print('-------------------------Score Manager-------------------------')
print('If you want a manual, type the following command: "command"')
scoredb = readScoreDB()
print()
print('-------File contents-------')
showScoreDB(scoredb, 'Name')  #처음 데이터 파일의 상태를 파악하기위해...
doScoreDB(scoredb)
writeScoreDB(scoredb)
```

Before >>>>> After

수정 방향: 사용자가 좀 더 편하게(?) 이용할 수 있도록 수정함

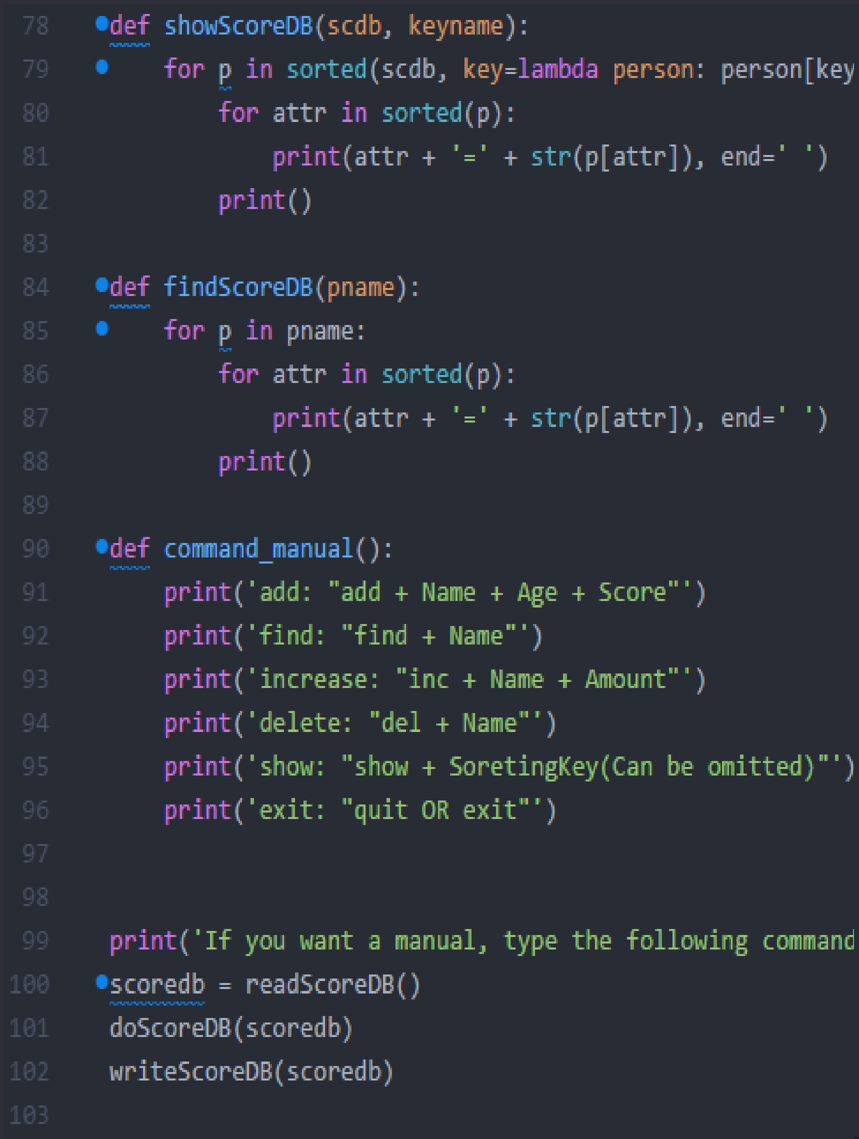
(진짜 사용자가 이 프로그램을 사용한다면 어떨까 라는 생각해봄)

Before >>>>> After

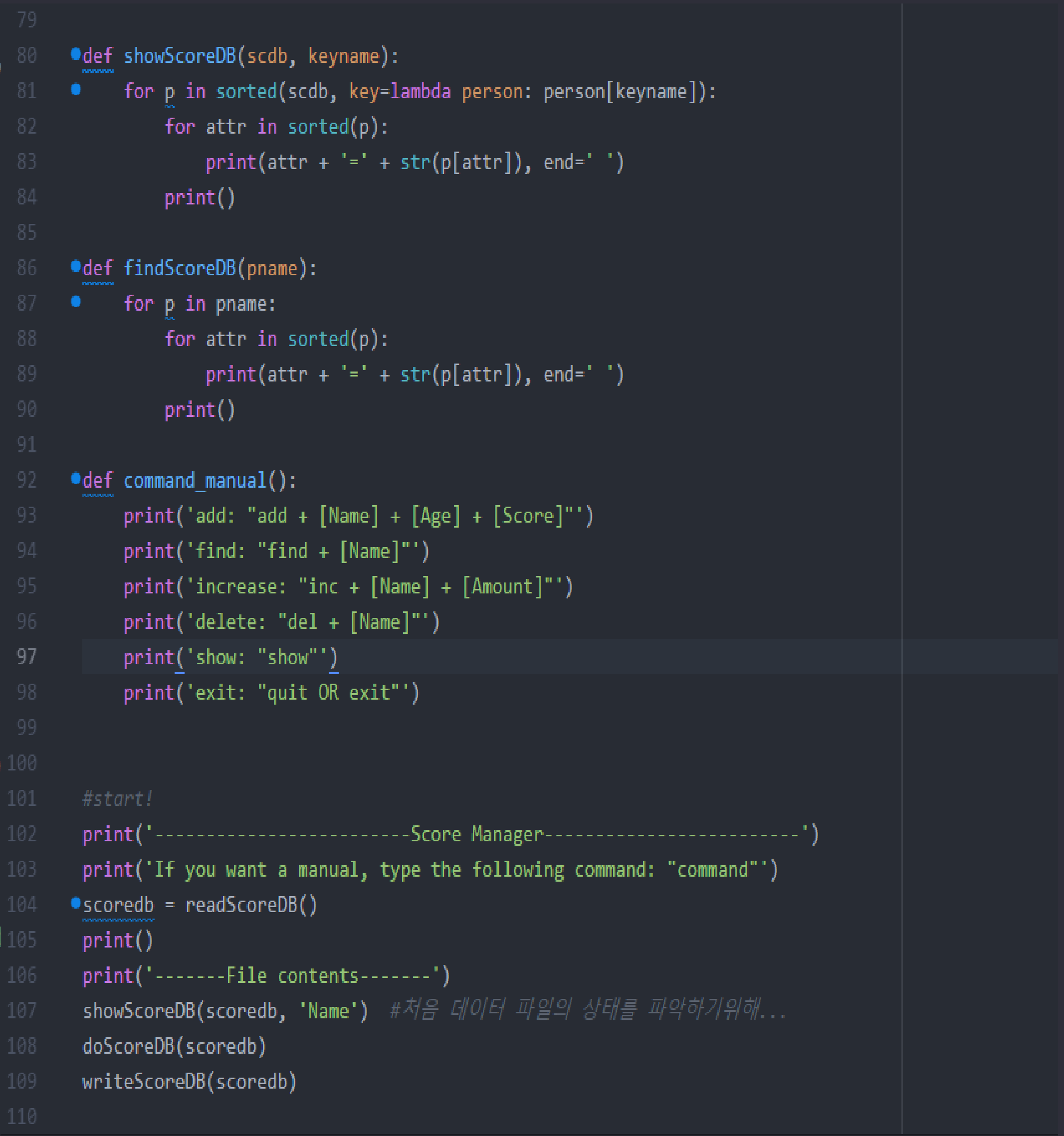
Cord Review

(Before > After)

사용자 편의(커맨드 매뉴얼 변경, 프로그램 시작 시 깔끔한(?) 화면)



```
def showScoreDB(scdb, keyname):
    for p in sorted(scdb, key=lambda person: person[key
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()

def findScoreDB(pname):
    for p in pname:
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()

def command_manual():
    print('add: "add + Name + Age + Score"')
    print('find: "find + Name"')
    print('increase: "inc + Name + Amount"')
    print('delete: "del + Name"')
    print('show: "show + SoretingKey(Can be omitted)"')
    print('exit: "quit OR exit"')


print('If you want a manual, type the following command
scoredb = readScoreDB()
doScoreDB(scoredb)
writeScoreDB(scoredb)
```



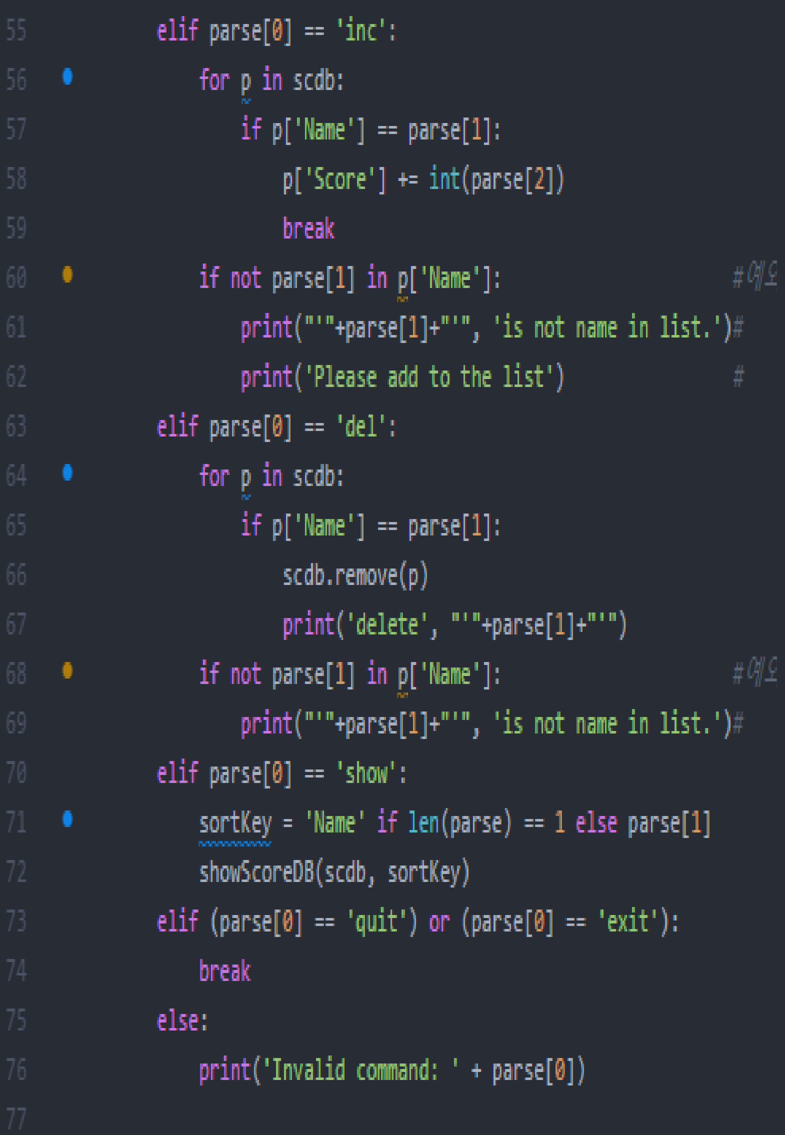
```

def showScoreDB(scdb, keyname):
    for p in sorted(scdb, key=lambda person: person[keyname]):
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()

def findScoreDB(pname):
    for p in pname:
        for attr in sorted(p):
            print(attr + '=' + str(p[attr]), end=' ')
        print()

def command_manual():
    print('add: "add + [Name] + [Age] + [Score]"')
    print('find: "find + [Name]"')
    print('increase: "inc + [Name] + [Amount]"')
    print('delete: "del + [Name]"')
    print('show: "show"')
    print('exit: "quit OR exit"')


#start!
print('------------------------Score Manager------------------------')
print('If you want a manual, type the following command: "command"')
scoredb = readScoreDB()
print()
print('-------File contents-------')
showScoreDB(scoredb, 'Name')  #처음 데이터 파일의 상태를 파악하기위해...
doScoreDB(scoredb)
writeScoreDB(scoredb)

```

Before >>>>> After

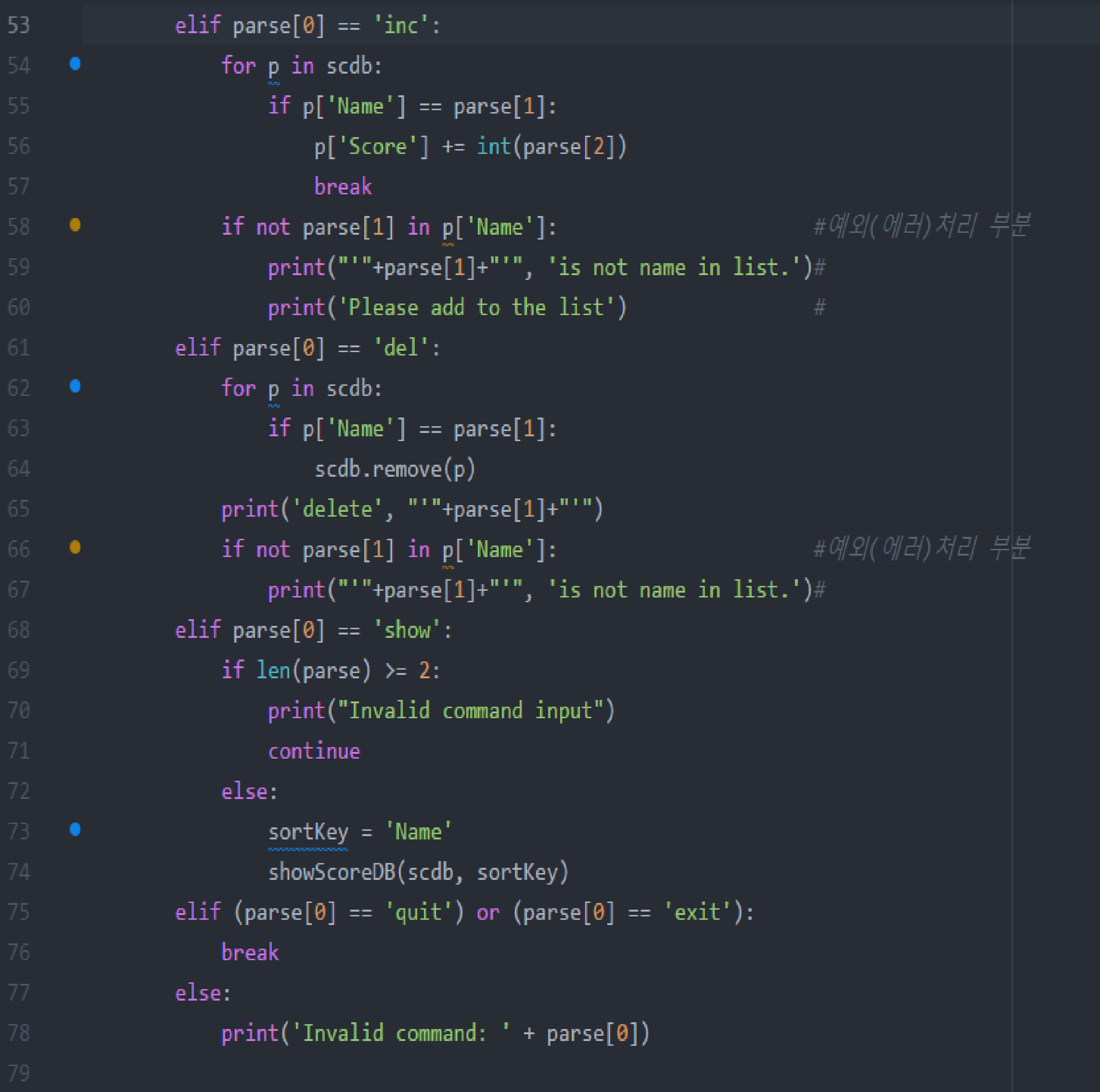
Cord Review

(Before > After)

예외처리 코드 수정('show' 커맨드, del 안내문 중복 출력 수정 등)



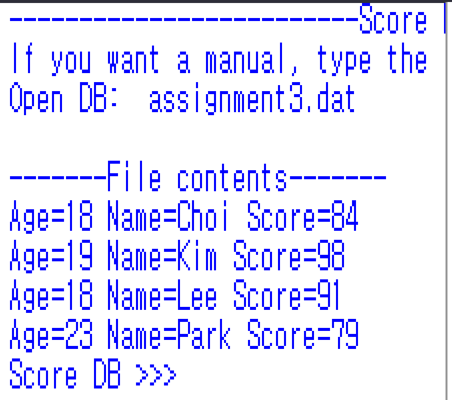
```
        elif parse[0] == 'inc':
            for p in scdb:
                if p['Name'] == parse[1]:
                    p['Score'] += int(parse[2])
                    break
            if not parse[1] in p['Name']:                      #예오
                print("'"+parse[1]+"'", 'is not name in list.')#
                print('Please add to the list')                 #
        elif parse[0] == 'del':
            for p in scdb:
                if p['Name'] == parse[1]:
                    scdb.remove(p)
                    print('delete', "'"+parse[1]+"'")
            if not parse[1] in p['Name']:                      #예오
                print("'"+parse[1]+"'", 'is not name in list.')#
        elif parse[0] == 'show':
            sortKey = 'Name' if len(parse) == 1 else parse[1]
            showScoreDB(scdb, sortKey)
        elif (parse[0] == 'quit') or (parse[0] == 'exit'):
            break
        else:
            print('Invalid command: ' + parse[0])
```



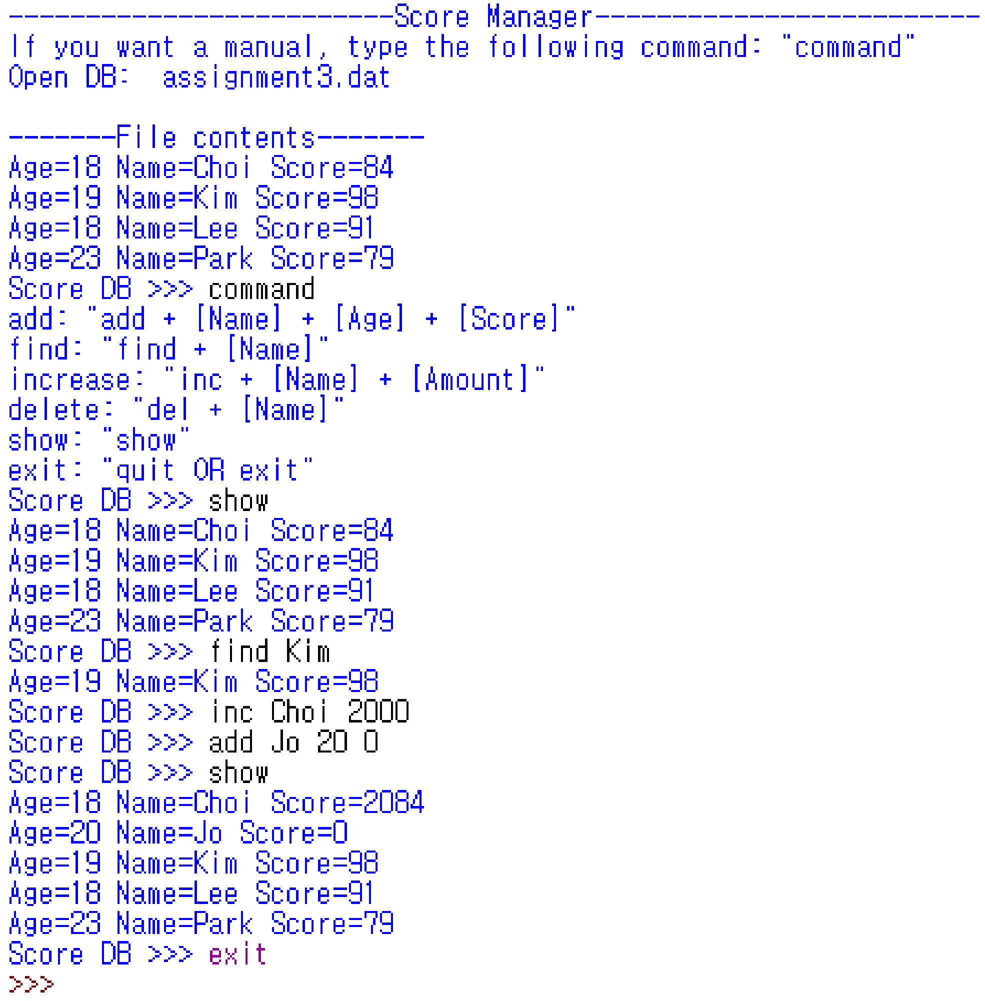
```
        elif parse[0] == 'inc':
            for p in scdb:
                if p['Name'] == parse[1]:
                    p['Score'] += int(parse[2])
                    break
            if not parse[1] in p['Name']:                         #예외(에러)처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
                print('Please add to the list')                 #
        elif parse[0] == 'del':
            for p in scdb:
                if p['Name'] == parse[1]:
                    scdb.remove(p)
            print('delete', "'"+parse[1]+"'")
            if not parse[1] in p['Name']:                         #예외(에러)처리 부분
                print("'"+parse[1]+"'", 'is not name in list.')#
        elif parse[0] == 'show':
            if len(parse) >= 2:
                print("Invalid command input")
                continue
            else:
                sortKey = 'Name'
                showScoreDB(scdb, sortKey)
        elif (parse[0] == 'quit') or (parse[0] == 'exit'):
            break
        else:
            print('Invalid command: ' + parse[0])
```

실행결과 화면

Cord Review
(Result)



```
-------------------------Score
If you want a manual, type the
Open DB:  assignment3.dat

-------File contents-------
Age=18 Name=Choi Score=84
Age=19 Name=Kim Score=98
Age=18 Name=Lee Score=91
Age=23 Name=Park Score=79
Score DB >>>
```



```
--------------------------Score Manager--------------------------
If you want a manual, type the following command: "command"
Open DB:  assignment3.dat

-------File contents-------
Age=18 Name=Choi Score=84
Age=19 Name=Kim Score=98
Age=18 Name=Lee Score=91
Age=23 Name=Park Score=79
Score DB >>> command
add: "add + [Name] + [Age] + [Score]"
find: "find + [Name]"
increase: "inc + [Name] + [Amount]"
delete: "del + [Name]"
show: "show"
exit: "quit OR exit"
Score DB >>> show
Age=18 Name=Choi Score=84
Age=19 Name=Kim Score=98
Age=18 Name=Lee Score=91
Age=23 Name=Park Score=79
Score DB >>> find Kim
Age=19 Name=Kim Score=98
Score DB >>> inc Choi 2000
Score DB >>> add Jo 20 0
Score DB >>> show
Age=18 Name=Choi Score=2084
Age=20 Name=Jo Score=0
Age=19 Name=Kim Score=98
Age=18 Name=Lee Score=91
Age=23 Name=Park Score=79
Score DB >>> exit
>>>
```

감사합니다!