Richard Hayes Crowley

03/15/2021

CSC\_157\_Lab\_07

**SOURCE CODE:**

class Student:

Scores = {}

*# initializing the constructor method*

def \_\_init\_\_(*self*, *name*, *grade*):

self.name = name

self.grade = grade

def getScores(*self*):

*# begin function to obtain scores #*

answer\_key = []

*# read into answer\_key list, the answer key from file*

answer\_key = [line.strip() *for* line *in* open("answers.txt", 'r')]

student\_answers = []

*# read into student\_answers list, student answers from file*

student\_answers = \

[line.strip().split(',') *for* line *in* open("data.txt", 'r')]

total\_score = 100

*for* student *in* student\_answers:

*if* self.name in student:

*for* idx, el *in* enumerate(student[1:]):

*if* el != answer\_key[idx]:

total\_score -= 10

Student.Scores[self.getName()] = total\_score

def getName(*self*):

*return* self.name

@ staticmethod

def sortDict():

*return* sorted(Student.Scores.items())

student\_objs = [

Student("Sammy Student", 65),

Student("Betty Sanchez", 45),

Student("Alice Brown", 100),

Student("Tom Schulz", 50),

Student("Richard Hayes Crowley", 0)

]

*for* index *in* range(len(student\_objs)):

student\_objs[index].getScores()

sortList = Student.sortDict()

*for* k, v *in* sortList:

print(k, "has score:", v)

**OUTPUT:**Graphical user interface, text, application

Description automatically generated