بسمه تعالی

دانشگاه آزاد اسلامی واحد زنجان

سؤالات امتحانی پایان ترم نیمسال دوم ۱۴۰۲-۱۴۰۱ - دانشکده فنی

نام درس : **برنامه سازی پیشرفته** نام استاد: مجتبی اعجمی تاریخ و ساعت امتحان: ۱۴۰۲/۰۳/۲۷ ساعت ۱۰:۳۰ مدت امتحان: ۷۵ دقیقه

نام و نام خانوادگی دانشجو: ..... شماره دانشجویی: .....

۱- فرض کنید یک فایل متنی به نام student.txt دارید که حاوی اطلاعاتی در مورد دانشجویان مختلف و نمرات آنهاست. هر خط در فایل دارای فرمت زیر است:

```
student_name,course1_grade,course2_grade,course3_grade
```

به عنوان مثال، یک خط می تواند به صورت زیر باشد:

```
Alice,85,90,95
```

یک برنامه پایتون بنویسید که فایل را بخواند و لیستی از اشیاء دانشجویی ایجاد کند. هر شی دانشجو باید از نوع کلاسی به نام Student باشد که دارای ویژگی ها و متدهای زیر باشد:

name	نام دانشجو
grades	لیست نمرات دانشجو
__init__(self, name, grades)	متد سازنده
__str__(self)	متدی که نام و نمرات دانشجو را به صورتی متنی می دهد
average(self)	متدی که معدل دانشجو را برمی گرداند
passed(self)	متدی که اگر معدل دانشجو بیشتر از ۶۰ باشد True برمیگرداند

پس از ایجاد لیست اشیاء دانشجویی، از درک لیست استفاده کنید تا یک لیست جدید ایجاد کنید که فقط شامل دانشجویانی باشد که هر سه درس را پاس کرده اند.

سپس، یک dictionary ایجاد کنید که نام هر درس را به فهرستی از اشیاء دانشجویان که آن درس را گذرانده‌اند، نگاشت کند. برای مثال، کلید "course1" باید به فهرستی از تمام اشیاء دانشجویانی که در لیست نمرات خود نمره ای برای "course1" دارند، نگاشت شود.

در نهایت تابعی به نام search_student بنویسید که دو پارامتر دارد: نام دانش آموز و دیکشنری. این تابع باید نام دانشجو را در دیکشنری جستجو کند و در صورت یافتن، یک رشته حاوی اطلاعات دانشجو یا اگر یافت نشد، "Not found" را برگرداند. برنامه باید Exception handling را انجام دهد.

موفق باشید

```python
# Define the Student class
class Student:
    def __init__(self, name, grades):
        self.name = name
       self.grades = grades
     def __str__(self):
        return f"{self.name} has an average grade of {self.average()}"

    def average(self):
        return sum(self.grades) / len(self.grades)
    def passed(self):
        return self.average() >= 60

# Create an empty list to store student objects
students = []
# Open the file and read each line
with open("students.txt") as file:
    for line in file:
        # Split the line by comma and strip any whitespace
        data = line.strip().split(",")
        # Get the name and grades from the data
        name = data[0]
        grades = [int(g) for g in data[1:]]
        # Create a student object and append it to the list
        student = Student(name, grades)
        students.append(student)

# Use list comprehension to create a new list of students who passed all three
courses
passed_students = [s for s in students if s.passed()]

# Create an empty dictionary to store course names and student lists
courses = {}

# Loop through each course name
for i in range(1, 4):
    course_name = f"course{i}"
    # Use list comprehension to create a list of students who took that course
    course_students = [s for s in students if len(s.grades) >= i]
    # Add the course name and student list to the dictionary
    courses[course_name] = course_students

# Define the search_student function
def search_student(name, dictionary):
    # Loop through each course name and student list in the dictionary
    for course_name, student_list in dictionary.items():
        # Loop through each student in the student list
        for student in student_list:
            # Check if the student name matches the parameter
            if student.name == name:
                # Return a string with the student's information
                return f"{student} took {course_name}."
    # If no match is found, return "Not found"
    return "Not found."
```