students = []

def add\_student(name):

students.append({"name": name,"grades":{}})

def remove\_student(name):

global students

students = [student for student in students if student["name"] != name]

def add\_grade(name, subject, grades):

for student in students:

if student["name"] == name:

student["grades"][subject] = grades

def calculate\_average(name):

for student in students:

if student["name"] == name:

grades = student["grades"]

return sum(grades) / len(grades)

return None

def mark\_attendance(name, attended):

for student in students:

if student["name"] == name:

if attended== True:

student["attendance"] = "present"

else:

student["attendance"] = "absent"

add\_student("Alice")

add\_student("Bob")

add\_grade("Alice", "Math", [95,70,85])

#print(calculate\_average("Alice"))

print(students)

mark\_attendance("Bob", True)

mark\_attendance("Alice", False)

print(students)

1. Code Complexity: How did you feel about extending the code? Was it straightforward? Did it feel like you were repeating code or introducing potential errors?
   * The code was fairly simple to work with in this example, though we had to change multiple functions to complete some requirements.
2. Data Management: How is the data (students, grades, attendance) managed? Are there any risks associated with directly modifying global variables?
   * Since the list is global, this means it can be accessed by external functions in other code, which is very low security.
3. Maintainability: How easy do you think it will be to maintain and further extend this code if the requirements change?
   * Maintaining and extending the code will be sort of a challenge. While in this example it was fairly simple, in more complicated programs, it would be very hard to maintain as functions are disrupted by other functions in the program.