

Python File Handling & Student Grades Assignment

Question 1: Grade Checker

Task: Take a score as input and print the grade.

◆ Code:

```
Score = int(input("Enter your score:"))
```

```
if score >= 90:
```

```
print("Grade: A")
```

```
elif score >= 80:
```

```
print("Grade: B")
```

```
elif score >= 70:
```

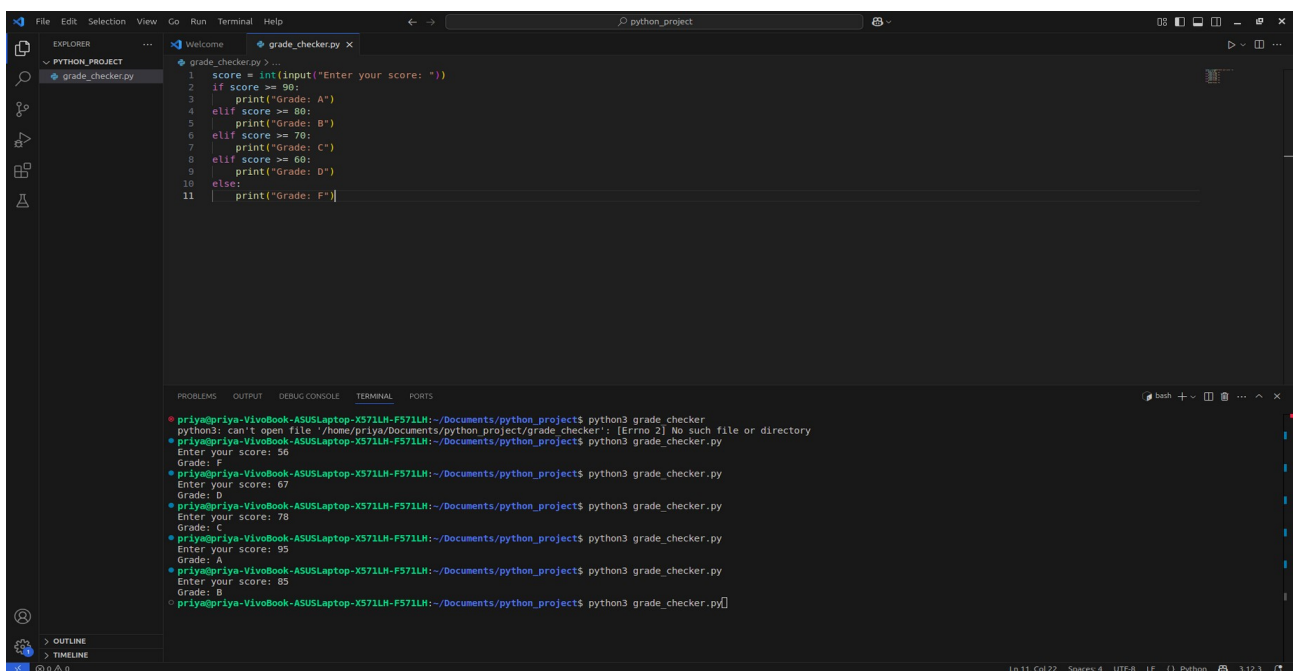
```
print("Grade: C")
```

```
elif score >= 60:
```

```
print("Grade: D")
```

```
else:
```

```
print("Grade: F")
```



The screenshot shows a Visual Studio Code editor with a Python file named `grade_checker.py`. The code implements a grade checker using conditional statements. Below the editor, the terminal window shows the execution of the script with various input scores and the corresponding output grades.

```
1 score = int(input("Enter your score: "))
2 if score >= 90:
3     print("Grade: A")
4 elif score >= 80:
5     print("Grade: B")
6 elif score >= 70:
7     print("Grade: C")
8 elif score >= 60:
9     print("Grade: D")
10 else:
11     print("Grade: F")
```

Terminal Output:

```
priya@priya-VivoBook-ASUSLaptop-X571LH-F571LH: ~/Documents/python_project$ python3 grade_checker.py
python3: can't open file '/home/priya/Documents/python_project/grade_checker': [Errno 2] No such file or directory
priya@priya-VivoBook-ASUSLaptop-X571LH-F571LH: ~/Documents/python_project$ python3 grade_checker.py
Enter your score: 50
Grade: F
priya@priya-VivoBook-ASUSLaptop-X571LH-F571LH: ~/Documents/python_project$ python3 grade_checker.py
Enter your score: 67
Grade: D
priya@priya-VivoBook-ASUSLaptop-X571LH-F571LH: ~/Documents/python_project$ python3 grade_checker.py
Enter your score: 78
Grade: C
priya@priya-VivoBook-ASUSLaptop-X571LH-F571LH: ~/Documents/python_project$ python3 grade_checker.py
Enter your score: 95
Grade: A
priya@priya-VivoBook-ASUSLaptop-X571LH-F571LH: ~/Documents/python_project$ python3 grade_checker.py
Enter your score: 85
Grade: B
priya@priya-VivoBook-ASUSLaptop-X571LH-F571LH: ~/Documents/python_project$ python3 grade_checker.py
```

Explanation :

Checks the input score using `if-elif-else` statements and prints a corresponding grade.

Question 2: Student Grades Using Dictionary

Task: Add, update, and display student grades.

◆ **Code:**

```
grades = {}

while True:

    print("\n1 . Add a new student")

    print("2 . Update an existing student's grade")

    print("3 . View all students' grades")

    print("4 . exit")

    choice = input("Enter your choice(1-4):")

    if choice == "1":

        name = input("Enter student's name: ")

        grade = input("Enter student's grade: ") # Use a different variable

        grades[name] = grade # Store the grade in the dictionary

    elif choice == "2":

        name = input("Enter student's name to update: ")

        if name in grades:

            new_grade = input("Enter new grade for {}: ".format(name))

            grades[name] = new_grade

        else:

            print("Student not found.")

    elif choice == "3":

        if grades:

            print("\nStudent Grades:")

            for student, grade in grades.items():

                print("{}: {}".format(student, grade))

        else:

            print("No students found.")

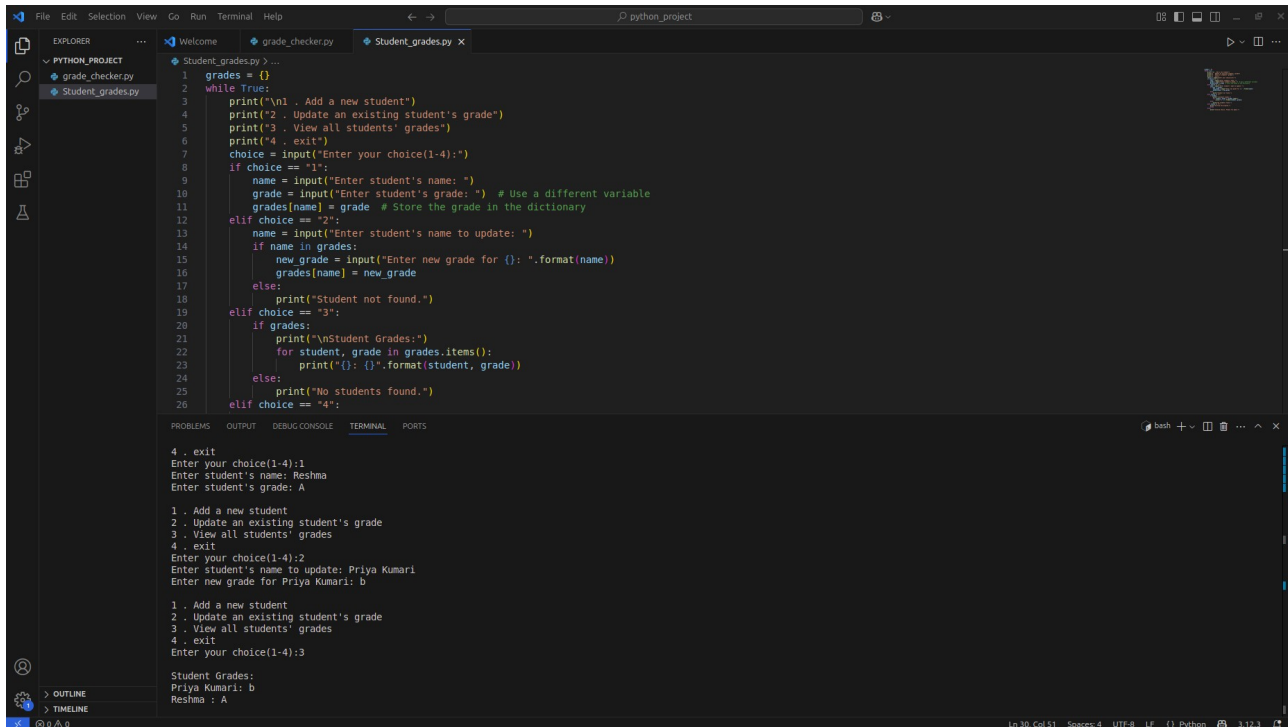
    elif choice == "4":

        print("Exiting the program.")
```

break

else:

print("Invalid choice. Please try again.")



```
1 grades = {}
2 while True:
3     print("\n1 . Add a new student")
4     print("2 . Update an existing student's grade")
5     print("3 . View all students' grades")
6     print("4 . exit")
7     choice = input("Enter your choice(1-4):")
8     if choice == "1":
9         name = input("Enter student's name: ")
10        grade = input("Enter student's grade: ") # Use a different variable
11        grades[name] = grade # Store the grade in the dictionary
12    elif choice == "2":
13        name = input("Enter student's name to update: ")
14        if name in grades:
15            new_grade = input("Enter new grade for {}: ".format(name))
16            grades[name] = new_grade
17        else:
18            print("Student not found.")
19    elif choice == "3":
20        if grades:
21            print("\nStudent Grades:")
22            for student, grade in grades.items():
23                print("{}: {}".format(student, grade))
24        else:
25            print("No students found.")
26    elif choice == "4":
27        break
28    else:
29        print("Invalid choice. Please try again.")
30    print()
```

4 . exit
Enter your choice(1-4):1
Enter student's name: Reshma
Enter student's grade: A

1 . Add a new student
2 . Update an existing student's grade
3 . View all students' grades
4 . exit
Enter your choice(1-4):2
Enter student's name to update: Priya Kumari
Enter new grade for Priya Kumari: b

1 . Add a new student
2 . Update an existing student's grade
3 . View all students' grades
4 . exit
Enter your choice(1-4):3

Student Grades:
Priya Kumari: b
Reshma : A

Explanation :

Uses a dictionary `grades` to store student names and their grades.

- Provides a menu to add, update, or view grades.
- Uses a loop to keep the menu running until "Exit" is chosen.

Question 3: Write to a File

Task: Create a file and write content into it.

◆ Code:

with open("myfile.txt", "w") as f:

f.write("This is my first file written using Python!\n")

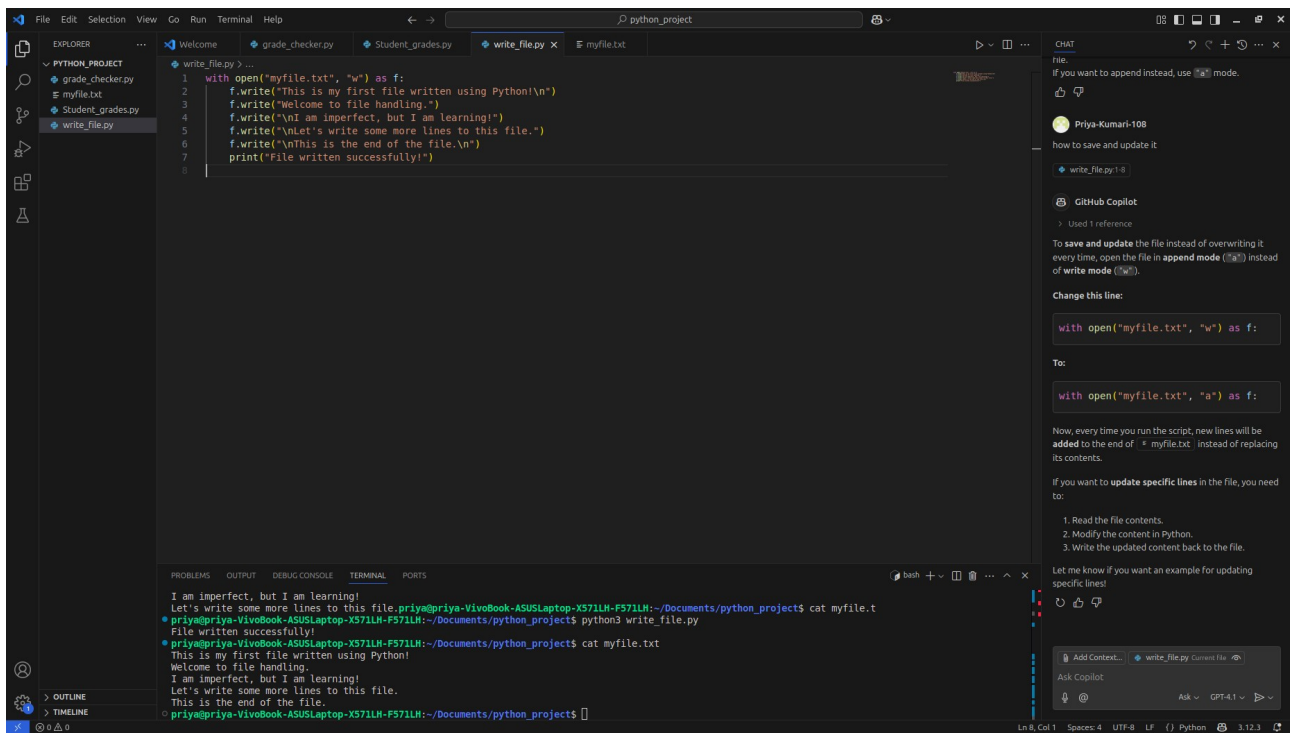
f.write("Welcome to file handling.")

f.write("\nI am imperfect, but I am learning!")

f.write("\nLet's write some more lines to this file.")

f.write("\nThis is the end of the file.\n")

print("File written successfully!")



Explanation :

`open("myfile.txt", "w")`: Opens or creates a file named `myfile.txt` in **write mode**.

- `file.write(...)`: Writes content to the file.
- `with` automatically closes the file when done.

Question 4: Read from the File

Task: Open the file created in Q3 and read its contents.

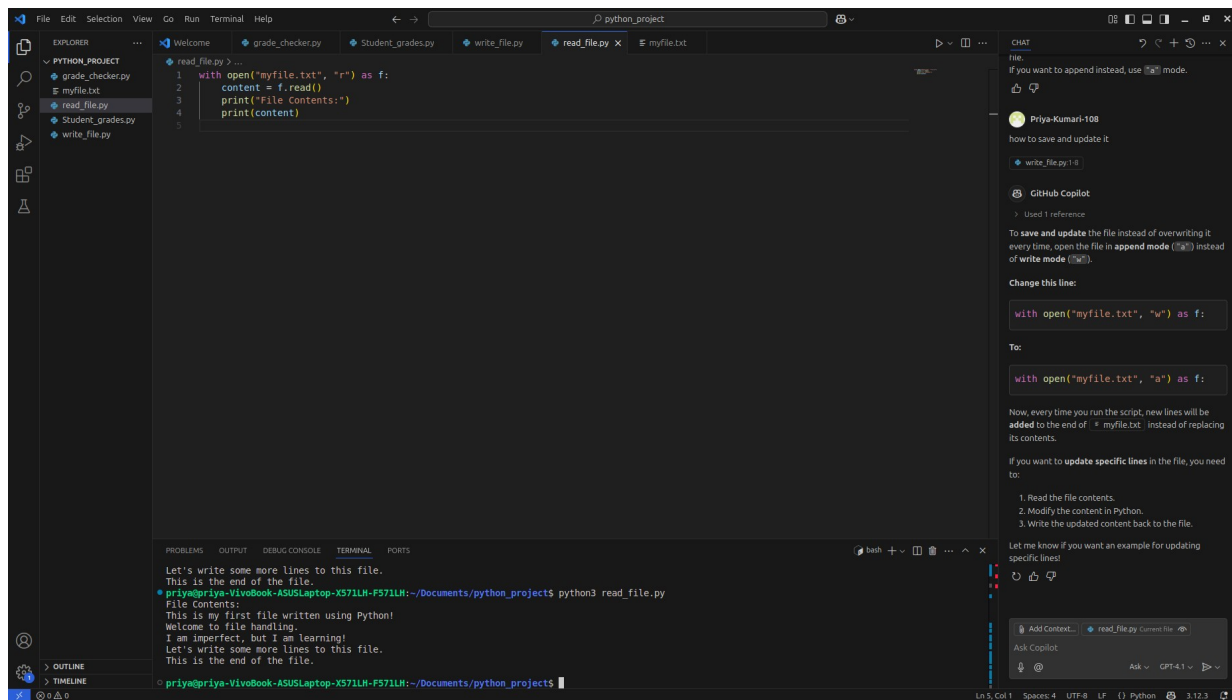
◆ Code:

```
with open("myfile.txt", "r") as f:
```

```
    content = f.read()
```

```
    print("File Contents:")
```

```
    print(content)
```



Explanation :

- `open("myfile.txt", "r")`: Opens the same file in **read mode**.
- `file.read()`: Reads the entire content of the file.
- `print(content)`: Displays the content in the terminal.