**Python Programs and Explanations**

**1.Grade Checker**

# Grade Checker Program

# Taking input from the user

score = int(input("Enter the score: "))

# Using if-else statements to determine grade

if score >= 90:

grade = "A"

elif score >= 80:

grade = "B"

elif score >= 70:

grade = "C"

elif score >= 60:

grade = "D"

else:

grade = "F"

# Printing the result

print(f"Grade: {grade}")

**EXPLANATION**

* We take an integer input for the score.
* The if-elif-else chain checks the range of the score and assigns a grade.
* The grade is printed using f-string formatting.

**2. Student Grades Dictionary**

# Student Grades Management

# Initial dictionary

student\_grades = {

"John": "A",

"Alice": "B",

"Mike": "C"

}

while True:

print("\n1. Add New Student")

print("2. Update Student Grade")

print("3. Print All Grades")

print("4. Exit")

choice = input("Enter your choice: ")

if choice == "1":

name = input("Enter student name: ")

grade = input("Enter student grade: ")

student\_grades[name] = grade

print(f"{name} added successfully!")

elif choice == "2":

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

if name in student\_grades:

grade = input("Enter new grade: ")

student\_grades[name] = grade

print(f"{name}'s grade updated successfully!")

else:

print("Student not found!")

elif choice == "3":

print("\n--- Student Grades ---")

for name, grade in student\_grades.items():

print(f"{name}: {grade}")

elif choice == "4":

print("Exiting...")

break

else

print("Invalid choice! Try again.")

**EXPLANATION**

* A dictionary stores student names and grades.
* User can add, update, and view grades.
* We use while True loop for a menu system, and dictionary operations (student\_grades[name] = grade) to modify data.

**3. Write to a File**

# Writing to a file

# Opening file in write mode

with open("example.txt", "w") as file:

file.write("Hello, this is a sample text file.\n")

file.write("This file is created using Python.\n")

print("Data written to 'example.txt' successfully.")

**Explanation:**

* open("filename", "w") opens a file in write mode.
* write() adds content to the file.
* with open(...) as file: ensures the file is automatically closed after writing.

**4. Read from a File**

# Reading from a file

# Opening file in read mode

with open("example.txt", "r") as file:

content = file.read()

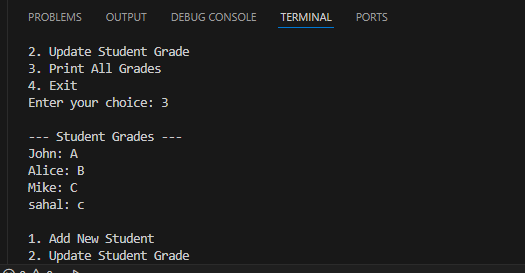
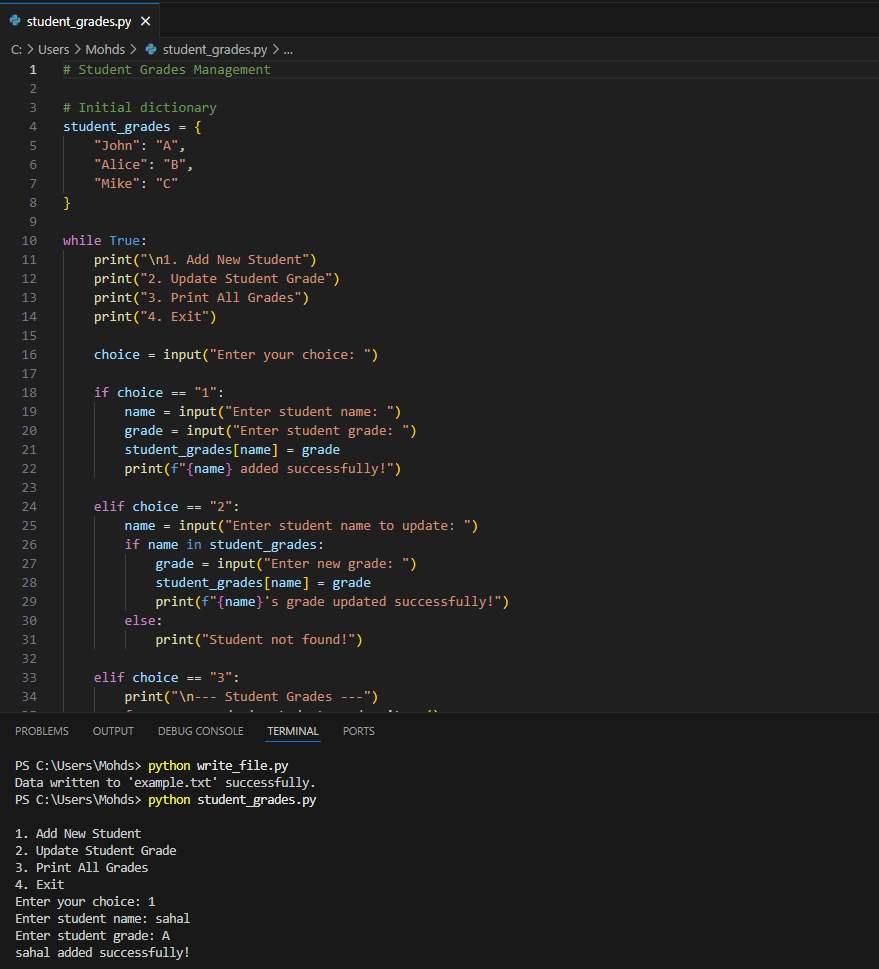
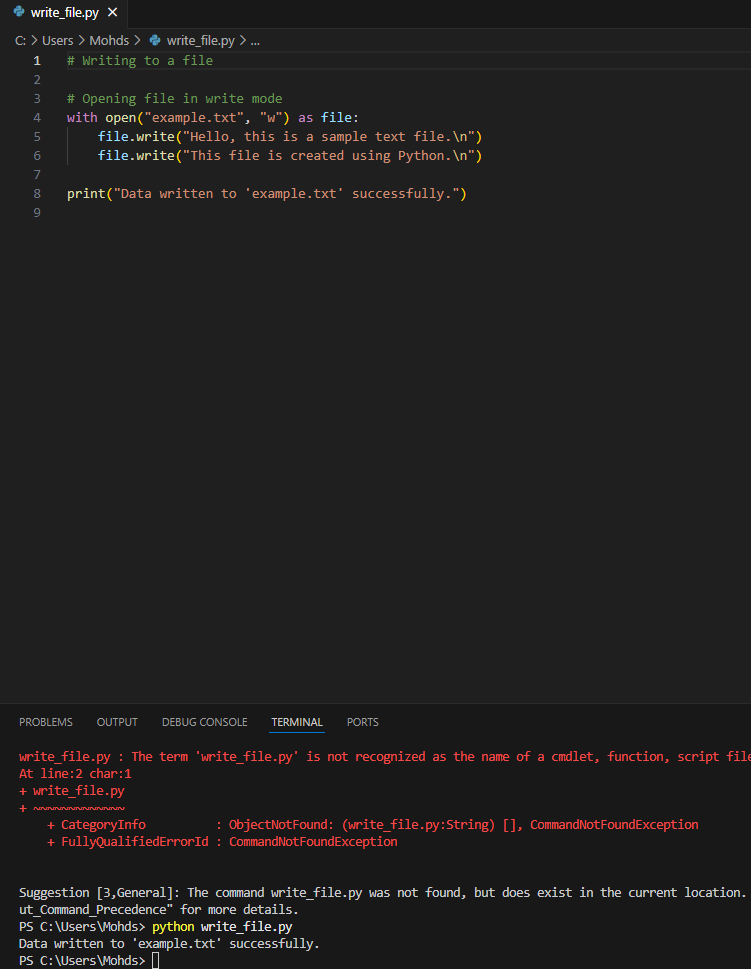
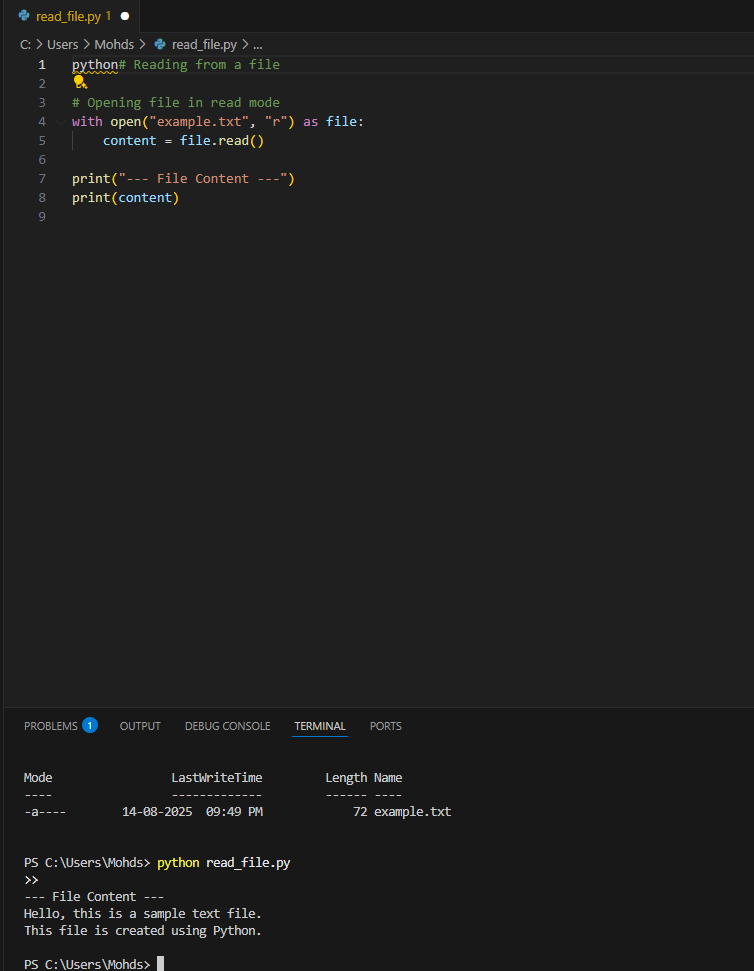
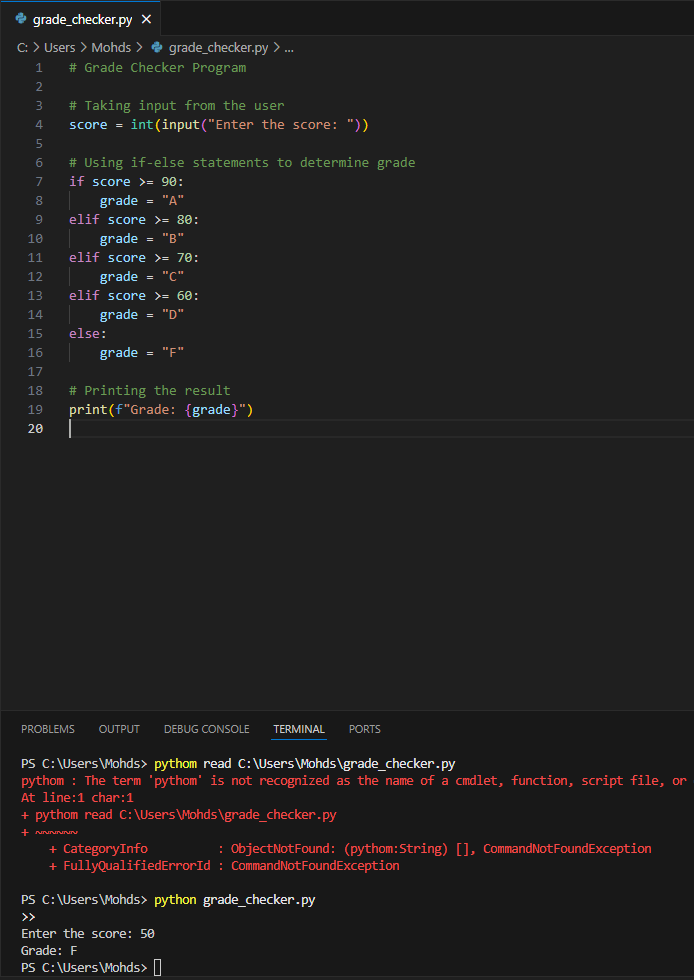
print("--- File Content ---")

print(content)

**Explanation:**

* open("filename", "r") opens a file in read mode.
* read() retrieves the content.
* The program prints the file content to the console.

**SCREENSHOTS**

****