Python Assignment

## Name: Jahnavi Dasari

## Course: Devops

## 1. Grade Checker

### Code

# Grade Checker  
score = int(input("Enter the 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")

### Explanation

- The program asks the user to enter a numeric score.  
- Using if-elif-else conditions, it checks the range of the score.  
- Based on the score, it prints the corresponding grade:  
 - 90 and above → A  
 - 80–89 → B  
 - 70–79 → C  
 - 60–69 → D  
 - Below 60 → F

### Screenshot

A black screen with white text

AI-generated content may be incorrect.

## 2. Student Grades Dictionary

### Code

# Student Grades Dictionary  
students = {  
 "Alice": "A",  
 "Bob": "B",  
 "Charlie": "C"  
}  
  
while True:  
 print("\nOptions: 1-Add 2-Update 3-View All 4-Exit")  
 choice = input("Enter choice: ")  
  
 if choice == "1":  
 name = input("Enter student name: ")  
 grade = input("Enter grade: ")  
 students[name] = grade  
 print("Student added successfully.")  
 elif choice == "2":  
 name = input("Enter student name to update: ")  
 if name in students:  
 grade = input("Enter new grade: ")  
 students[name] = grade  
 print("Grade updated.")  
 else:  
 print("Student not found.")  
 elif choice == "3":  
 print("\nStudent Grades:")  
 for name, grade in students.items():  
 print(name, ":", grade)  
 elif choice == "4":  
 break  
 else:  
 print("Invalid choice. Try again.")

### Explanation

- A dictionary is used to store student names and their grades.  
- Options are given to the user:  
 1. Add a new student and grade.  
 2. Update an existing student’s grade.  
 3. View all students and their grades.  
 4. Exit the program.  
- This demonstrates the use of dictionary operations and loops.

### Screenshot

A screenshot of a computer program

AI-generated content may be incorrect.

## 3. Write to a File

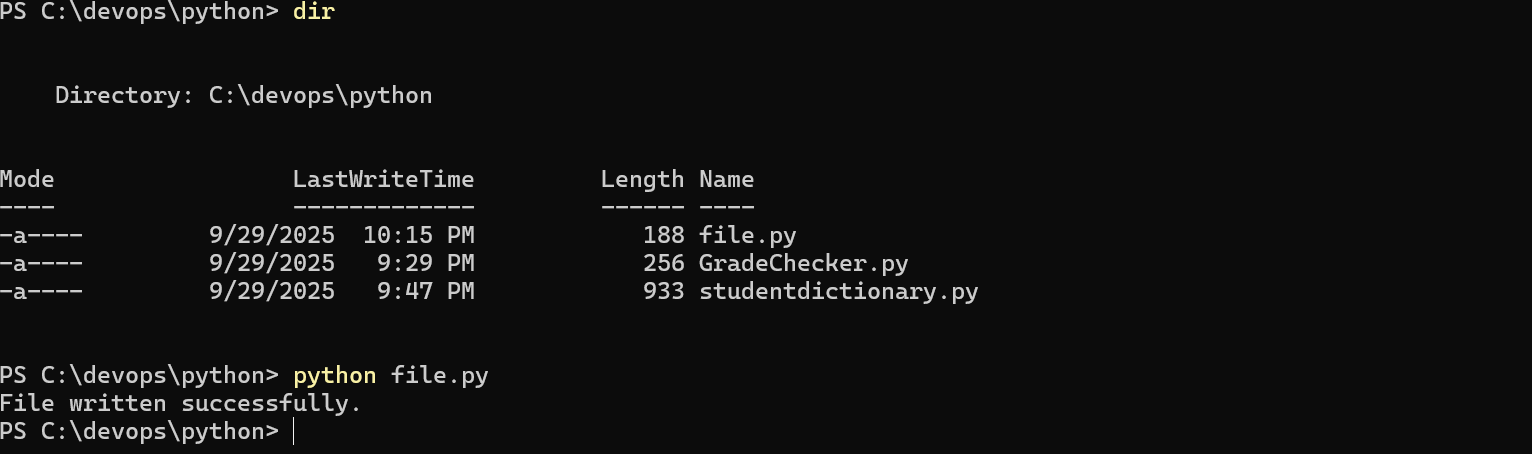
### Code

# Write to a File  
with open("output.txt", "w") as file:  
 file.write("Hello, this is a test file.\n")  
 file.write("This is the second line.")  
print("File written successfully.")

### Explanation

- The program opens a file called output.txt in write mode.  
- It writes two lines of text into the file using the write() function.  
- The file is automatically saved and closed because of the with statement.  
- Finally, it prints a confirmation message.

### Screenshot



## 4. Read from a File

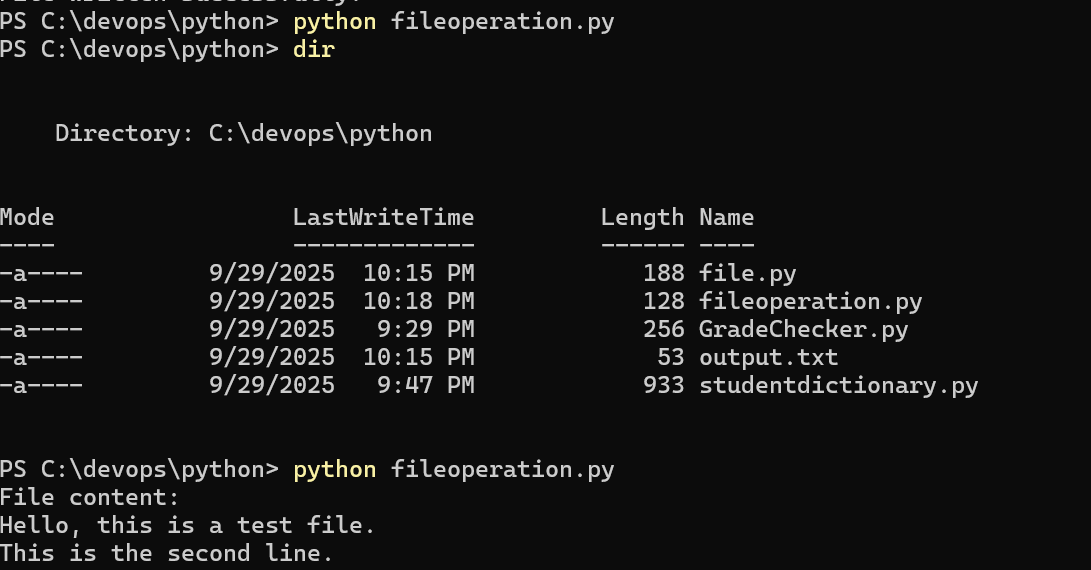
### Code

# Read from a File  
with open("output.txt", "r") as file:  
 content = file.read()  
  
print("File content:")  
print(content)

### Explanation

- The program opens output.txt in read mode.  
- It reads the entire content of the file using the read() function.  
- The content is printed to the screen.  
- This confirms that the file created earlier can be successfully read.

### Screenshot



## Conclusion

In this assignment, I’ve successfully implemented:  
1. Conditional statements to check grades.  
2. Dictionary operations to manage student data.  
3. File handling operations (write and read) to work with text files.  
  
This covers fundamental concepts of Python programming, including decision-making, data structures, and file handling.