**Assignment 2**

1= Example Explanation:

If the score is >= 90, grade is "A".

If the score is between 80–89, grade is "B".

If the score is between 70–79, grade is "C".

If the score is between 60–69, grade is "D".

If the score is below 60, grade is "F".

Concepts Used: if-else conditional statements.

**---------------------Code -----**

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

if score >= 90:

print("Grade: A")

elif score >= 80:

print("Grade: B")

elif score >= 70:

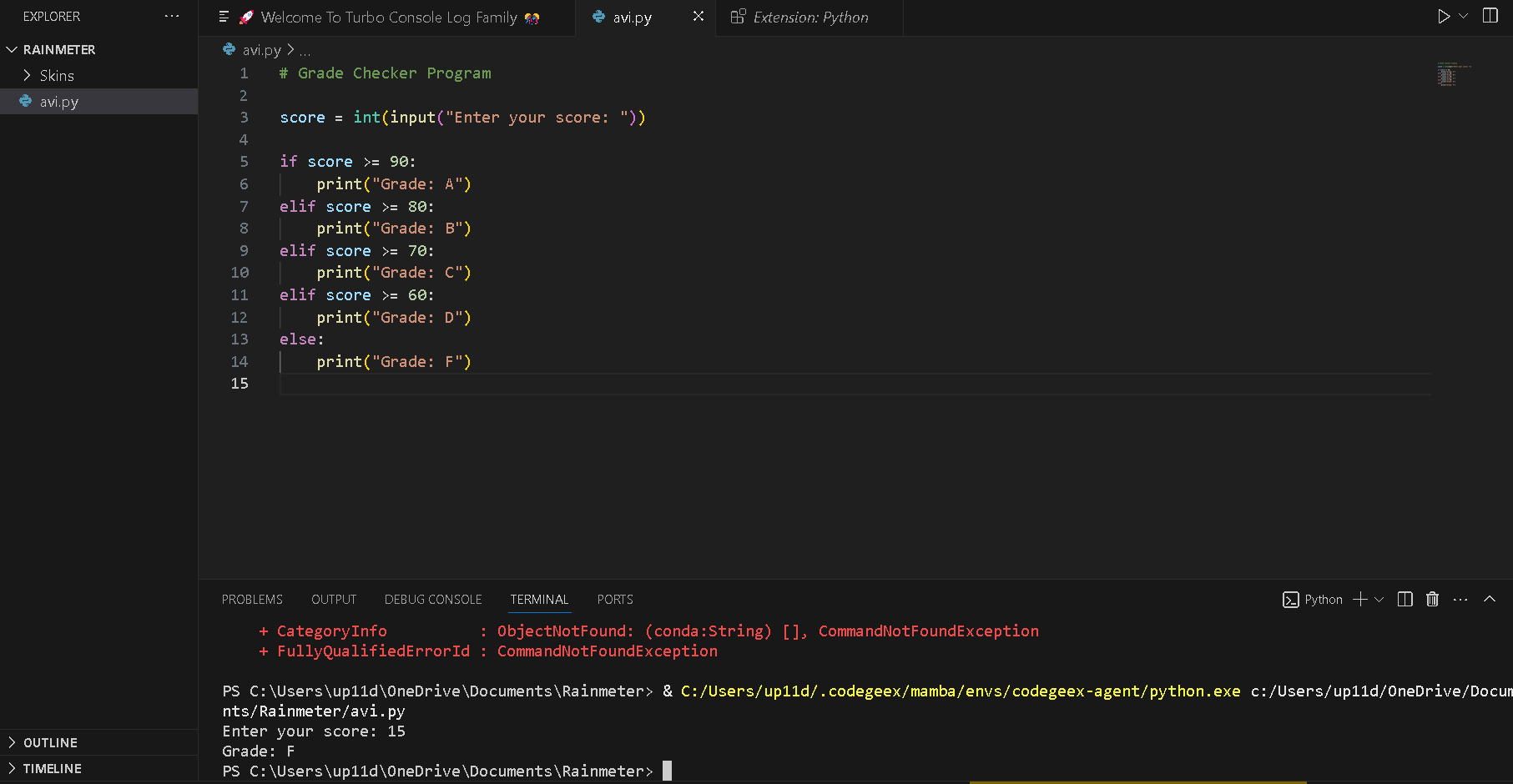
print("Grade: C")

elif score >= 60:

print("Grade: D")

else:

print("Grade: F")



2. Student Grades (Dictionary Operations)

Task: Create a dictionary where keys are student names and values are their grades.

Operations Performed:

Add a new student and grade.

Update an existing student’s grade.

Print all student grades.

Example Explanation:

A dictionary is created like:

grades = {"Aman": "B", "Neha": "A"}

New students can be added using grades["Rahul"] = "C".

Existing grades can be updated by reassigning the value.

The full dictionary is displayed using a loop or print(grades).

Concepts Used: Dictionary creation, updating values, if-else checks.

**---------------------Code -----**

# Student Grades Dictionary

grades = {} # empty dictionary

while True:

print("\n1. Add Student")

print("2. Update Student Grade")

print("3. Print All Grades")

print("4. Exit")

choice = int(input("Enter your choice: "))

if choice == 1:

name = input("Enter student name: ")

grade = input("Enter student grade: ")

grades[name] = grade

print(f"Added {name} with grade {grade}.")

elif choice == 2:

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

if name in grades:

grade = input("Enter new grade: ")

grades[name] = grade

print(f"Updated {name}'s grade to {grade}.")

else:

print("Student not found.")

elif choice == 3:

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

for student, grade in grades.items():

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

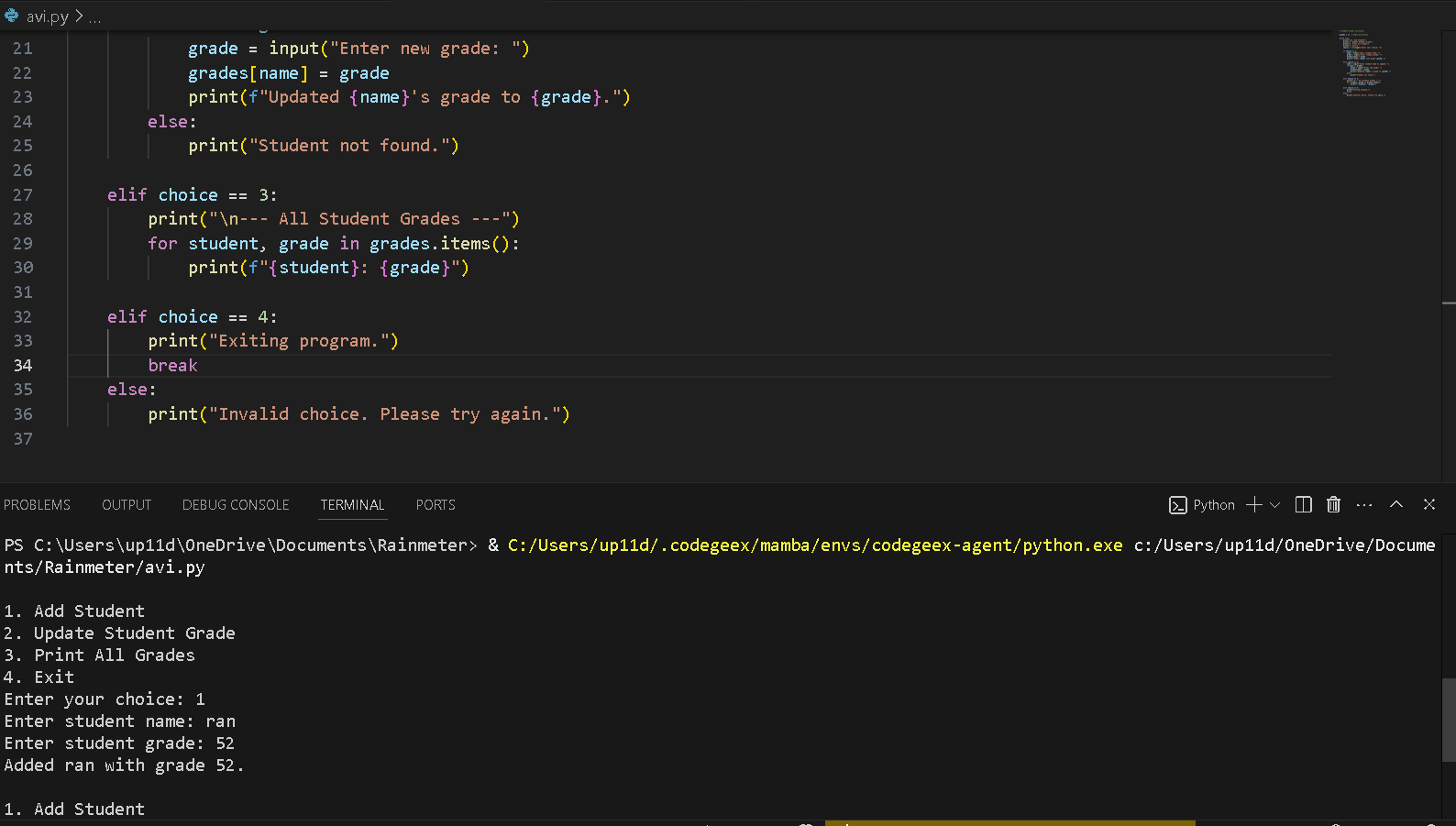
elif choice == 4:

print("Exiting program.")

break

else:

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



3. Write to a File

Task: Create a text file and write some content into it.

Method Used:

Open the file in write ("w") mode.

Use file.write() to write content.

Close the file with file.close() (or use with open(...)).

Example Explanation:

file = open("example.txt", "w")

file.write("This is my first file writing program.")

file.close()

Concepts Used: File handling functions (open, write, close).

**---------------------Code -----**

# Writing to a File

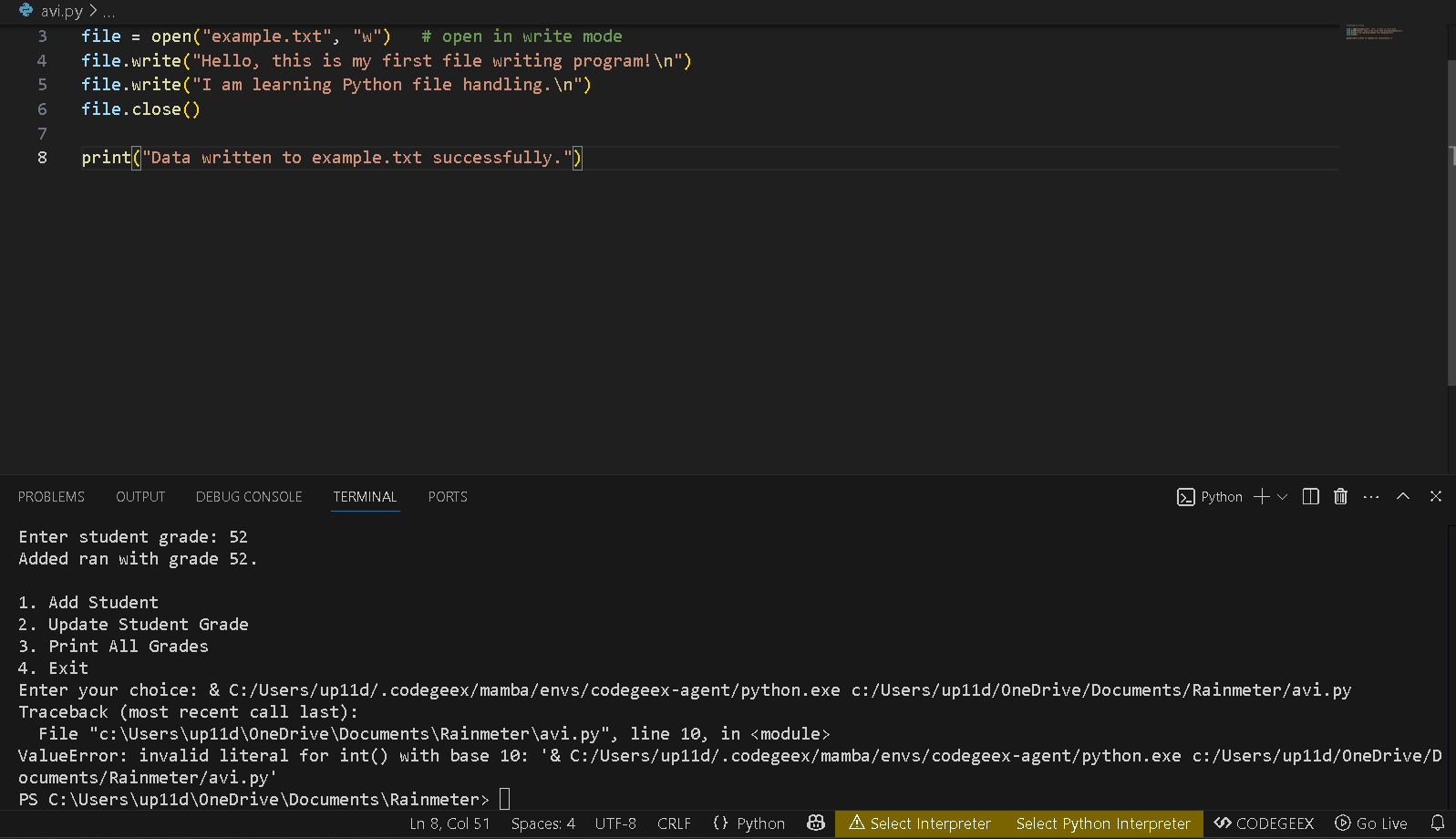
file = open("example.txt", "w") # open in write mode

file.write("Hello, this is my first file writing program!\n")

file.write("I am learning Python file handling.\n")

file.close()

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



4. Read from a File

Task: Open a file in read mode and display its content.

Method Used:

Open the file in read ("r") mode.

Use file.read() to read the contents.

Print the content on the screen.

Example Explanation:

file = open("example.txt", "r")

content = file.read()

print(content)

file.close()

Concepts Used: File reading (open, read, print, close).

**---------------------Code -----**

# Reading from a File

file = open("example.txt", "r") # open in read mode

content = file.read()

print("File Content:\n")

print(content)

file.close()

