

## 1. Grade Checker

Take a score as input and print the grade based on the following:

90+ : "A"

80-89 : "B"

70-79 : "C"

60-69 : "D"

Below 60 : "F"

here we used a basic if else statement to carry out marks and all

**Command**`grade=int(input("Enter your marks"))`

```
if (grade > 90):  
    print("A")  
  
elif (80<grade<89):  
    print("B")  
  
elif (70<grade<79):  
    print("D")  
  
elif (60<grade<69):  
    print("E")
```

```
else:  
  
    print("F")
```

## Input

40

## Result

F

## Screenshot

```
grade=int(input("Enter your marks"))  
if (grade > 90):  
    print("A")  
  
elif (80<grade<89):  
    print("B")  
  
elif (70<grade<79):  
    print("C")  
  
elif (60<grade<69):  
    print("D")  
  
else:  
    print("F")  
  
[8] ✓ 4.4s  
... F
```

## Explanation

If else blocks act like decision switches here. Gives decision on their number range

## 2 Student Grades

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

Allow the user to:

Add a new student and grade.

Update an existing student's grade.

Print all student grades.

Used dictionary and basic operations. Using if else:

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

Allow the user to:

Add a new student and grade.

Update an existing student's grade.

Print all student grades.

Used dictionary and basic operations. Using if else:

### Command

```
student_grades = {}

while True:

    print("\nOptions:")

    print("1. Add new student and grade")

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

    print("3. Print all student grades")

    print("4. Exit")
```

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

if choice == "1":

    name = input("Enter student name: ")

    grade = input("Enter student grade: ")

    if name in student_grades:

        print(f"{name} already exists.")

        student_grades[name] = grade

    else:

        student_grades.update({"name":name, "grade":grade})

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

elif choice == "2":

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

    if name in student_grades:

        new_grade = input("Enter new grade: ")

        student_grades[name] = new_grade

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

    else:

        print(f"{name} not found in records.")
```

```
elif choice == "3":

    print("\nStudent Grades:")

    for name, grade in student_grades.items():

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


elif choice == "4":

    print("Exiting program.")

    break


else:

    print("Invalid choice. Please enter 1-4.")
```

**Input:**

**1**

**Mohan**

**E**

**4**

**Result**

**Student Grades:**

**name: Mohan**

**grade: E**

## Screenshot

```
student_grades = {}

while True:
    print("\nOptions:")
    print("1. Add new student and grade")
    print("2. Update existing student's grade")
    print("3. Print all student grades")
    print("4. Exit")

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

    if choice == "1":
        name = input("Enter student name: ")
        grade = input("Enter student grade: ")

        if name in student_grades:
            print(f"{name} already exists.")
            student_grades[name] = grade
        else:
            student_grades.update({"name": name, "grade": grade})
            print(f"{name} added successfully.")

    elif choice == "2":
        name = input("Enter student name to update: ")
        if name in student_grades:
            new_grade = input("Enter new grade: ")
            student_grades[name] = new_grade
            print(f"{name}'s grade updated.")
        else:
            print(f"{name} not found in records.")

    elif choice == "3":
        print("\nStudent Grades:")
        for name, grade in student_grades.items():
            print(f"{name}: {grade}")

    elif choice == "4":
        print("Exiting program.")
        break

    else:
        print("Invalid choice. Please enter 1-4.")

(4) ✓ 17.0s
...
Options:
1. Add new student and grade
2. Update existing student's grade
3. Print all student grades
4. Exit
```

## Explanation

Initiates a blank dictionary and use operations 1 , 2, 3, 4 for adding, updating, printing , exiting

## 3. Write to a File

Write a program to create a text file and write some content to it.

## Command

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

    file.write("Name: John Doe\nGrade: A\n")

    file.write("Name: Jane Smith\nGrade: B+\n")

print("Data written to student_data.txt")
```

## Result

Data written to student\_data.txt

## Screenshot

```
with open("student_data.txt", "w") as file:
    file.write("Name: John Doe\nGrade: A\n")
    file.write("Name: Jane Smith\nGrade: B+\n")

print("Data written to student_data.txt")
```

[5] ✓ 0.0s

... Data written to student\_data.txt

## Explanation

Opens a file in write mode and use file.write operation to write the content

Using file functions like write and open.

## 4. Read from a File

We used open in read mode and file.read to read and print to display.

## Command

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

    content = file.read()

print("Contents of student_data.txt:")

print(content)
```

## Result

Contents of student\_data.txt:

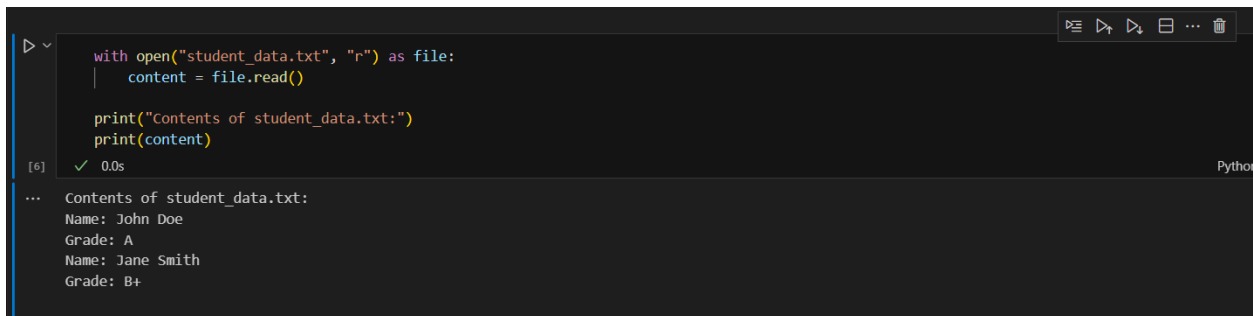
Name: John Doe

Grade: A

Name: Jane Smith

Grade: B+

## Screenshot



The screenshot shows a Python IDE with a dark theme. The top pane contains the following code:

```
with open("student_data.txt", "r") as file:  
    content = file.read()  
  
print("Contents of student_data.txt:")  
print(content)
```

The bottom pane shows the output of the code execution:

```
... Contents of student_data.txt:  
Name: John Doe  
Grade: A  
Name: Jane Smith  
Grade: B+
```

## Explanation

Open an existing file in read mode and use `file.read` to read the content to variable then prints the content

**Submission Guidelines -:** Attach Screenshots or command along with explanation and submit in doc(google doc or microsoft doc) format or share github link