

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.

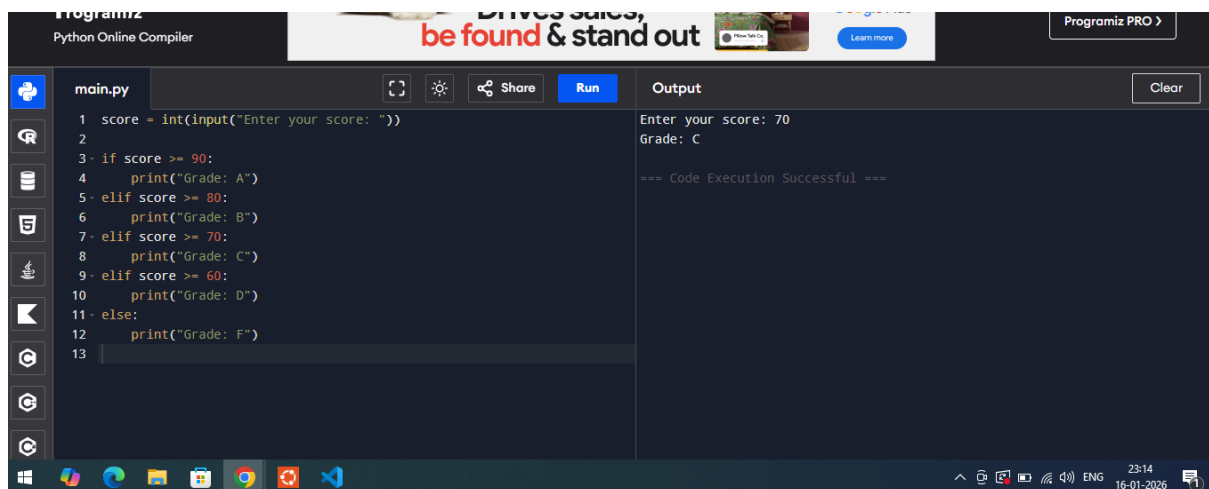
Ans :

User enters marks

Program checks conditions from top to bottom

First matching condition executes

Output grade is printed



The screenshot shows a web-based Python compiler interface. The code in the editor is as follows:

```
1 score = int(input("Enter your score: "))
2
3 if score >= 90:
4     print("Grade: A")
5 elif score >= 80:
6     print("Grade: B")
7 elif score >= 70:
8     print("Grade: C")
9 elif score >= 60:
10    print("Grade: D")
11 else:
12    print("Grade: F")
13
```

The output window shows the following text:

```
Enter your score: 70
Grade: C

=== Code Execution Successful ===
```

The interface includes a top banner with the text "be found & stand out", a "Run" button, and a "Clear" button. The bottom status bar shows the time as 23:14 on 16-01-2025.

2 Student Grades

Create a dictionary where the keys are student names and the values are their grades. Allow the user to:

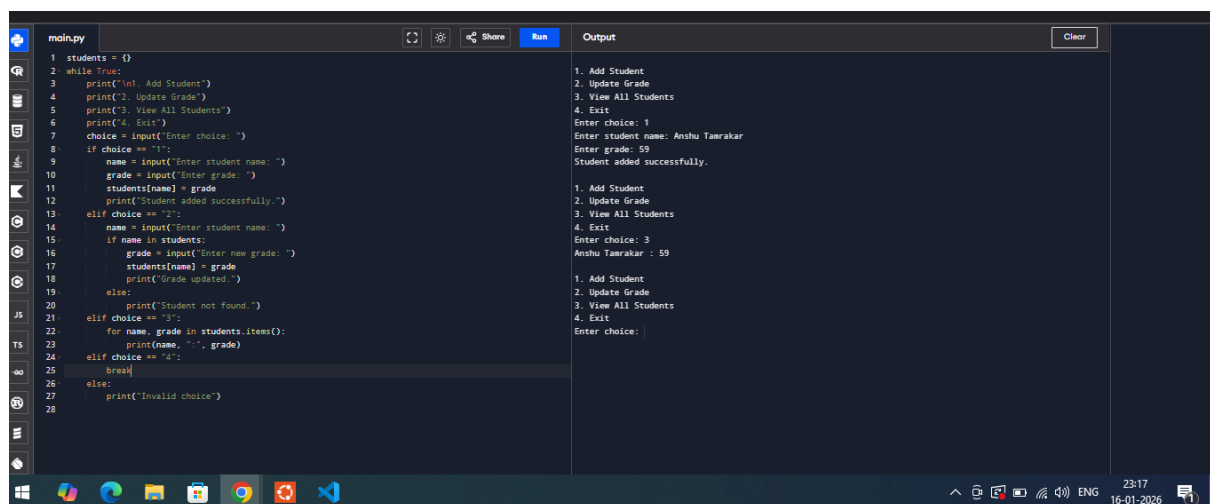
Add a new student and grade.

Update an existing student's grade.

Print all student grades.

Used dictionary and basic operations. Using if else:

Ans



```
main.py
1 students = {}
2 while True:
3     print("\n1. Add Student")
4     print("\n2. Update Grade")
5     print("\n3. View All Students")
6     print("\n4. Exit")
7     choice = input("Enter choice: ")
8     if choice == "1":
9         name = input("Enter student name: ")
10        grade = input("Enter grade: ")
11        students[name] = grade
12        print("Student added successfully.")
13    elif choice == "2":
14        name = input("Enter student name: ")
15        if name in students:
16            grade = input("Enter new grade: ")
17            students[name] = grade
18            print("Grade updated.")
19        else:
20            print("Student not found.")
21    elif choice == "3":
22        for name, grade in students.items():
23            print(name, "-", grade)
24    elif choice == "4":
25        break
26    else:
27        print("Invalid choice")
28
```

Output

```
1. Add Student
2. Update Grade
3. View All Students
4. Exit
Enter choice: 1
Enter student name: Anshu Tamrakar
Enter grade: 59
Student added successfully.

1. Add Student
2. Update Grade
3. View All Students
4. Exit
Enter choice: 3
Anshu Tamrakar : 59

1. Add Student
2. Update Grade
3. View All Students
4. Exit
Enter choice:
```

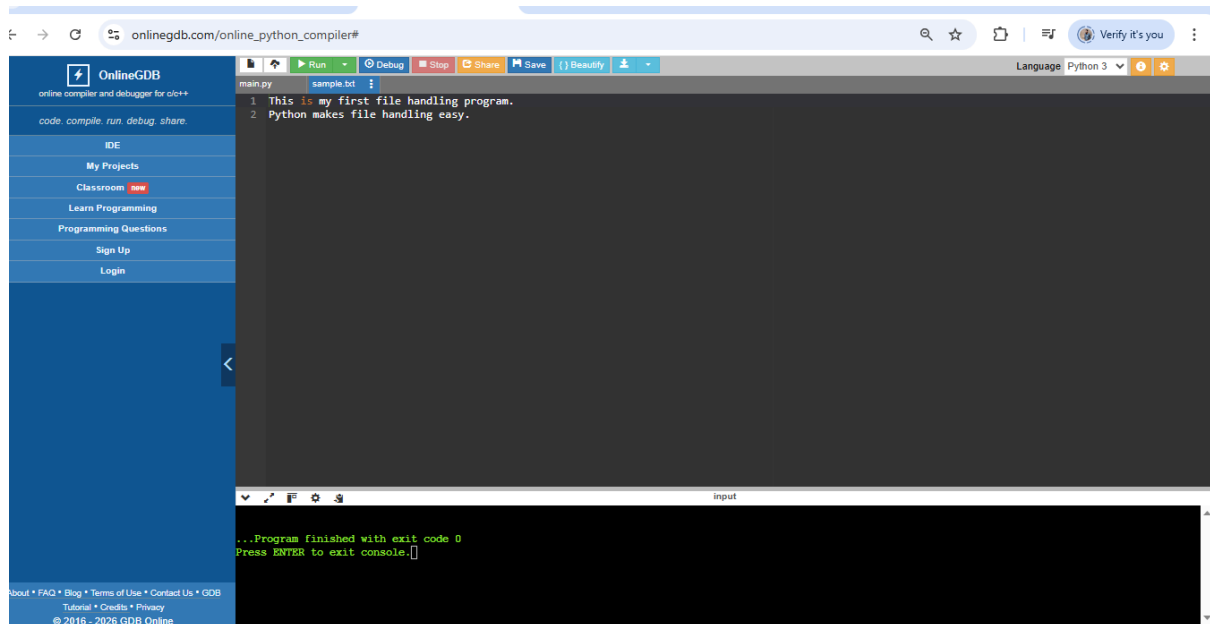
Dictionary stores student data Menu-driven program using while

If else handles operations in keyword checks existing students

3. Write to a File

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

Using file functions like write and open.



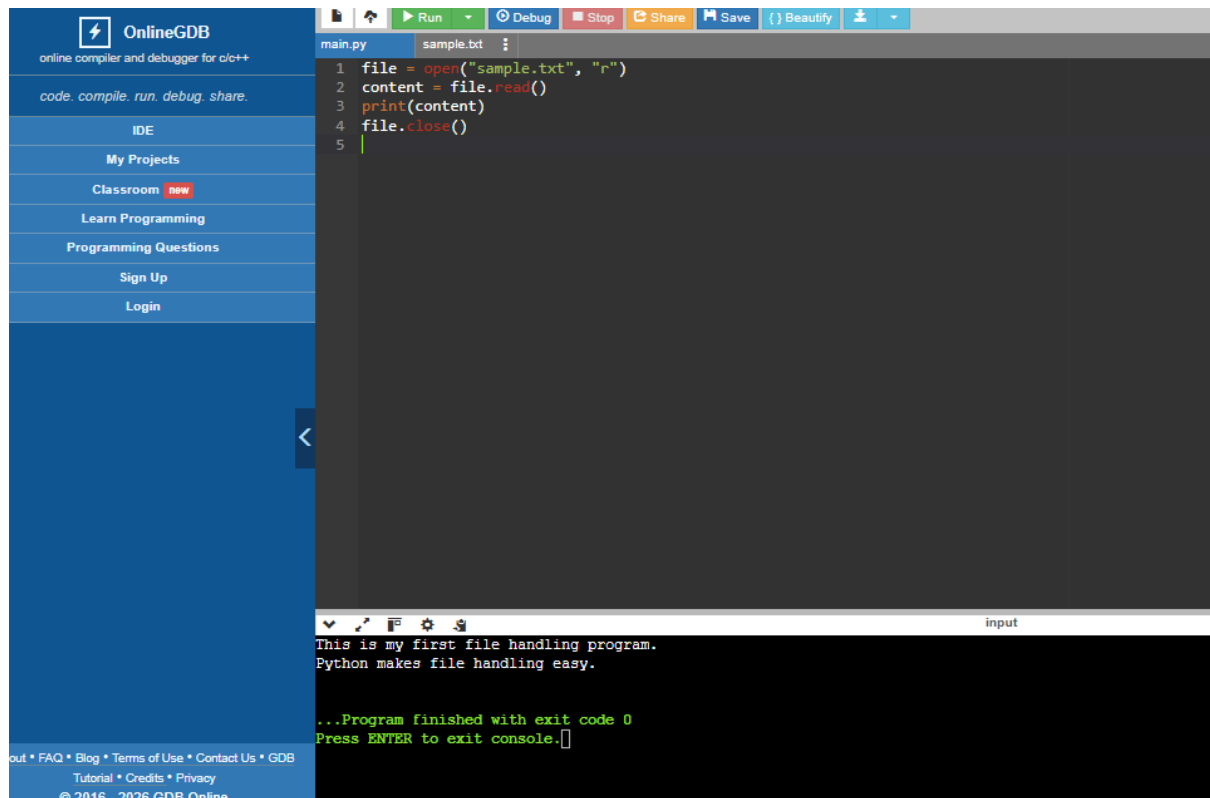
The program creates a text file named sample.txt and writes content into it using Python file handling.

The file was successfully created and the written content is visible. you can see in the right side of the [main.py](#) file.

4. Read from a File

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

Ans :



The screenshot shows the OnlineGDB web interface. On the left is a blue sidebar with navigation links: OnlineGDB, code, compile, run, debug, share, IDE, My Projects, Classroom (marked 'new'), Learn Programming, Programming Questions, Sign Up, and Login. The main area has a top toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. Below the toolbar, a file explorer shows 'main.py' and 'sample.txt'. The code editor displays a Python script:

```
1 file = open("sample.txt", "r")
2 content = file.read()
3 print(content)
4 file.close()
5
```

 At the bottom, a console window shows the program's output:

```
This is my first file handling program.
Python makes file handling easy.

...Program finished with exit code 0
Press ENTER to exit console.
```

The file sample.txt is opened in read mode ("r") file.read() reads the entire content of the file.

The content is printed on the screen using printf() file.close() closes the file after reading

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