Python Assignment - 1

1. Grade Checker

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

90+: "A"

80-89: "B"

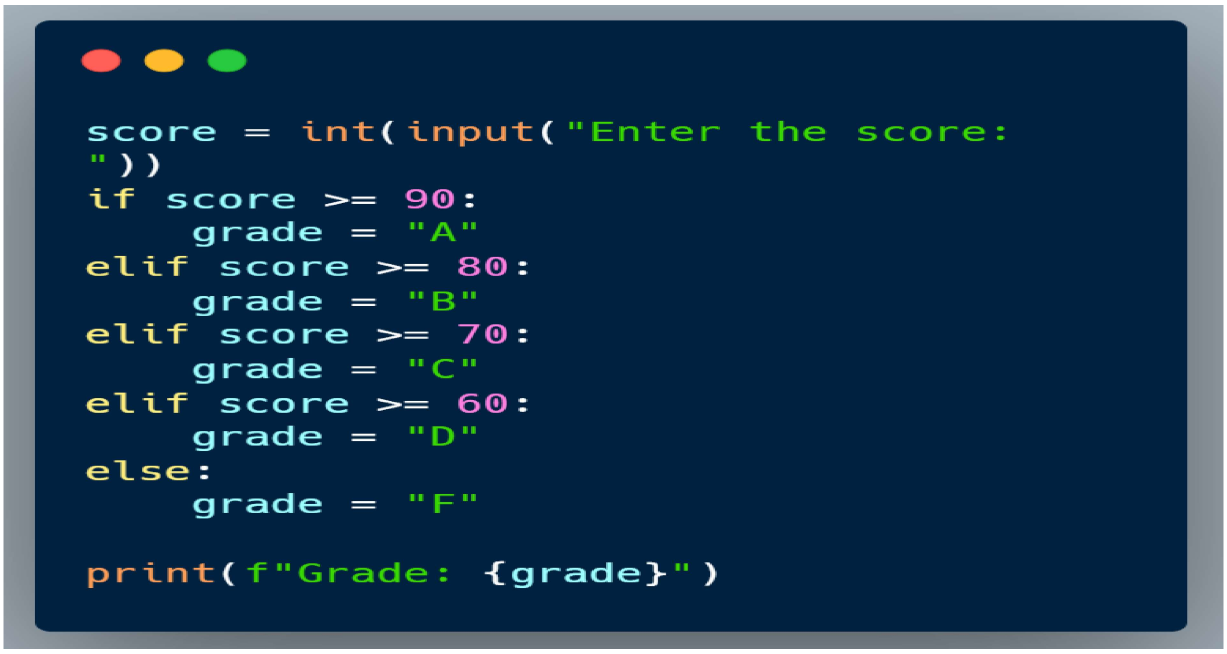
70-79: "C"

60-69: "D"

Below 60: "F"

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

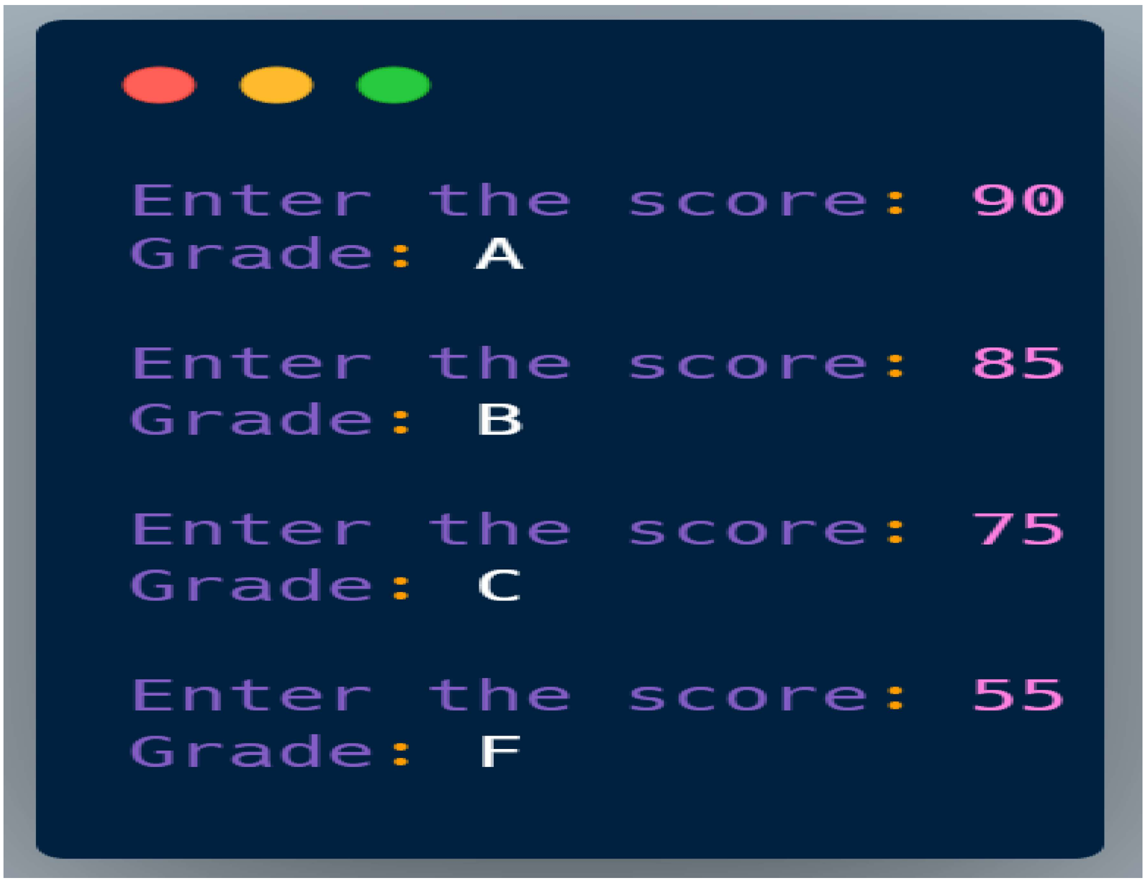
Code:



```python
score = int(input("Enter the score:
"))
if score >= 90:
    grade = "A"
elif score >= 80:
    grade = "B"
elif score >= 70:
    grade = "C"
elif score >= 60:
    grade = "D"
else:
    grade = "F"

print(f"Grade: {grade}")
```

Output:



```
Enter the score: 90
Grade: A

Enter the score: 85
Grade: B

Enter the score: 75
Grade: C

Enter the score: 55
Grade: F
```

Explanation:

- **input():** Prompts the user to enter something (in this case, a score).
- **int():** Converts the input string to an integer, since input is taken as text by default.
- The value is stored in the variable **score**.
- If the score is 90 or more, the grade is set to **"A"**.
- **elif** means "else if".
- If the first condition wasn't true and the score is between **80 and 89**, grade is **"B"**.
- If the score is between **70 and 79**, grade is **"C"**.
- If the score is between **60 and 69**, grade is **"D"**.
- If the score is **below 60**, grade is **"F"**.
- This prints the grade using an **f-string** for formatting.

2. Student Grades

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

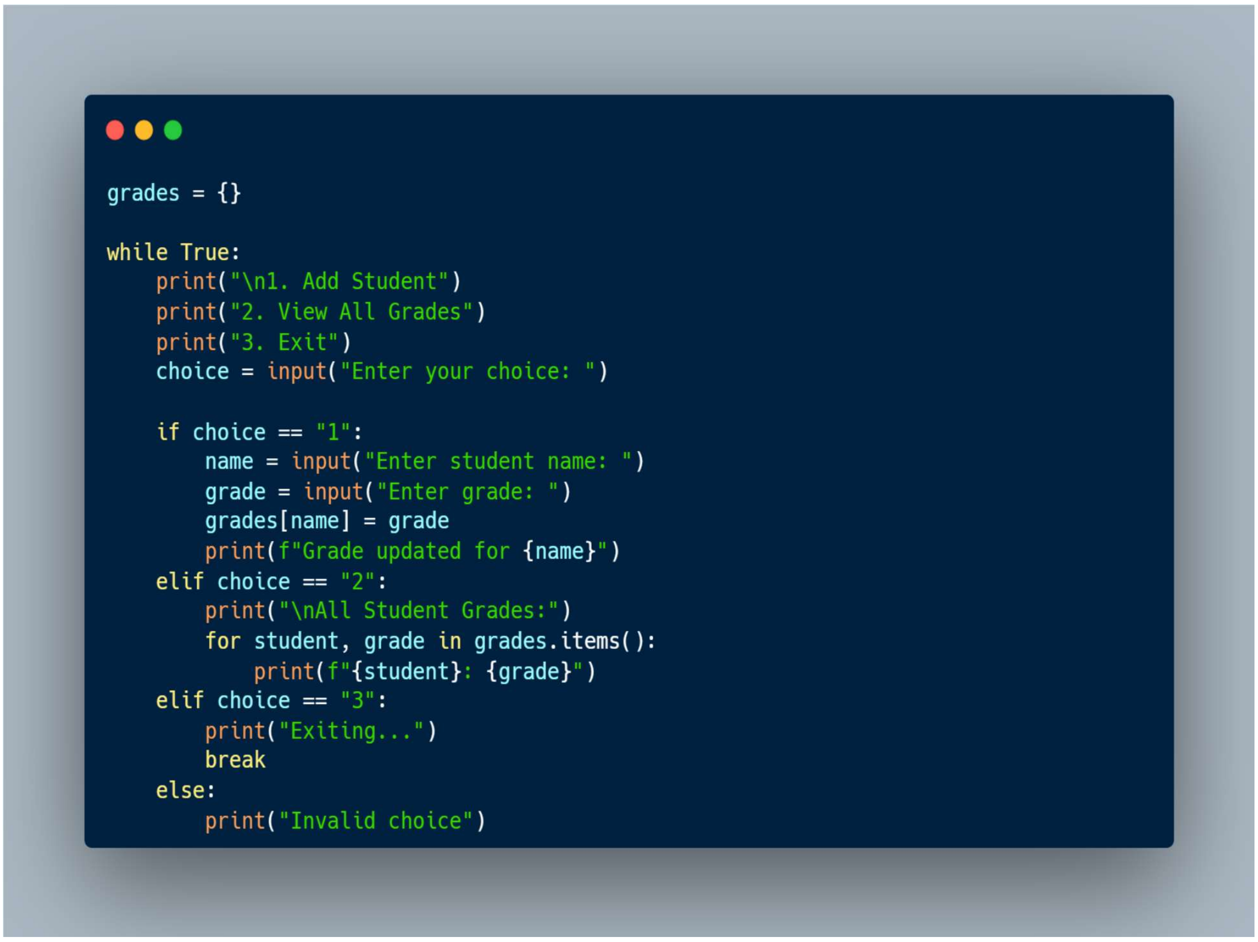
Add a new student and grade.

Update an existing student's grade.

Print all student grades.

Used a dictionary and basic operations. Using if else:

Code:

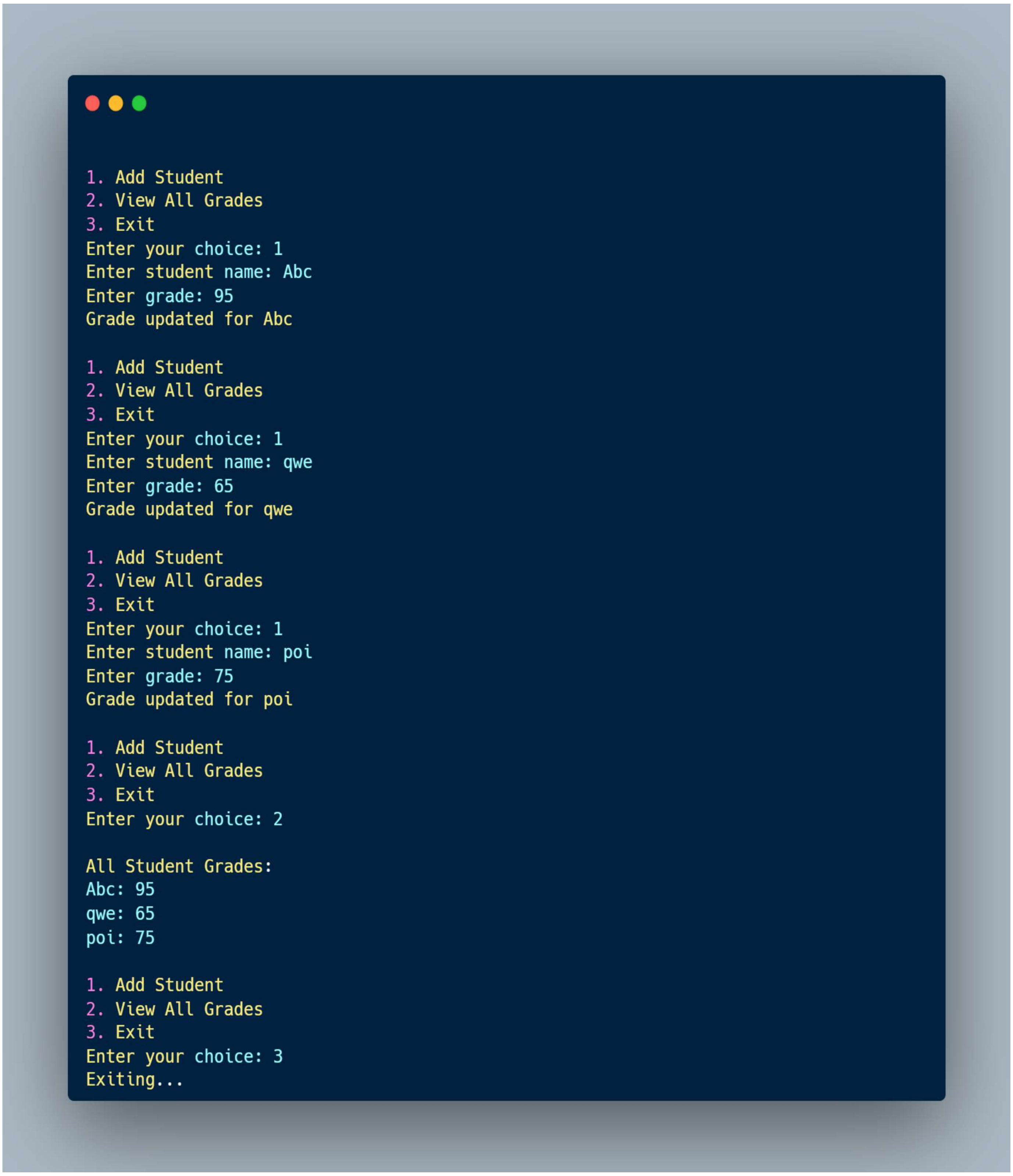


```python
grades = {}

while True:
    print("\n1. Add Student")
    print("2. View All Grades")
    print("3. Exit")
    choice = input("Enter your choice: ")

    if choice == "1":
        name = input("Enter student name: ")
        grade = input("Enter grade: ")
        grades[name] = grade
        print(f"Grade updated for {name}")
    elif choice == "2":
        print("\nAll Student Grades:")
        for student, grade in grades.items():
            print(f"{student}: {grade}")
    elif choice == "3":
        print("Exiting...")
        break
    else:
        print("Invalid choice")
```

Output:



```
1. Add Student
2. View All Grades
3. Exit
Enter your choice: 1
Enter student name: Abc
Enter grade: 95
Grade updated for Abc

1. Add Student
2. View All Grades
3. Exit
Enter your choice: 1
Enter student name: qwe
Enter grade: 65
Grade updated for qwe

1. Add Student
2. View All Grades
3. Exit
Enter your choice: 1
Enter student name: poi
Enter grade: 75
Grade updated for poi

1. Add Student
2. View All Grades
3. Exit
Enter your choice: 2

All Student Grades:
Abc: 95
qwe: 65
poi: 75

1. Add Student
2. View All Grades
3. Exit
Enter your choice: 3
Exiting...
```

Explanation:

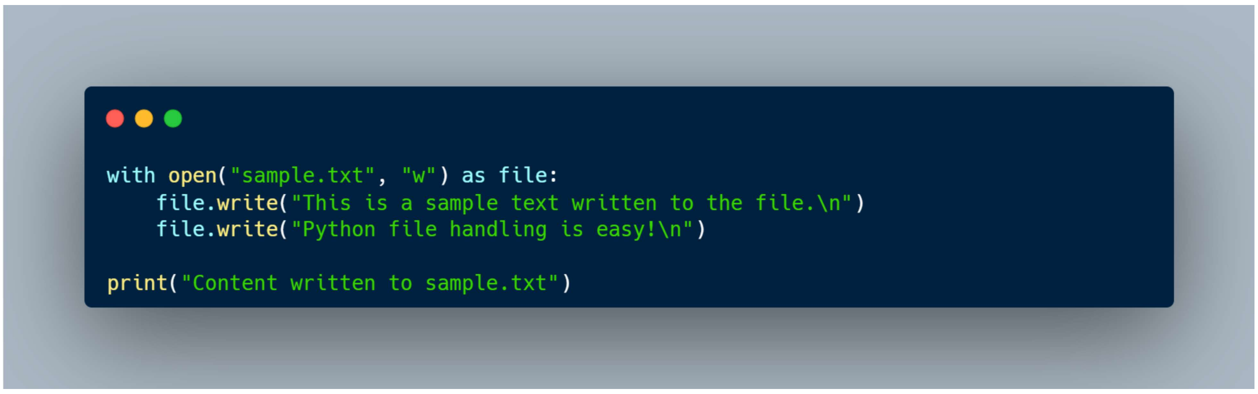
- It maintains a dictionary called grades where:
- **Key** = Student's name
- **Value** = Grade (like A, B, etc.)
- The user can:
    1. Add a student and their grade
    2. View all student grades
    3. Exit the program
- Initializes an empty dictionary to store student names and grades.
- Starts an infinite loop to keep showing the menu until the user chooses to exit.
- Displays the options available to the user.
- Takes the user’s choice as input.
- **Option 1: Add Student**
    - Prompts the user to enter a **student name** and their **grade.**
    - Adds or updates the dictionary entry with grades[name] = grade.
- **Option 2: View All Grades**
    - Prints all students and their corresponding grades stored in the dictionary using **for** loop.
- **Option 3: Exit**
    - Exits the loop and terminates the program.
- **Invalid Choice**
    - Displays a warning if the user enters an option other than 1, 2, or 3.

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.

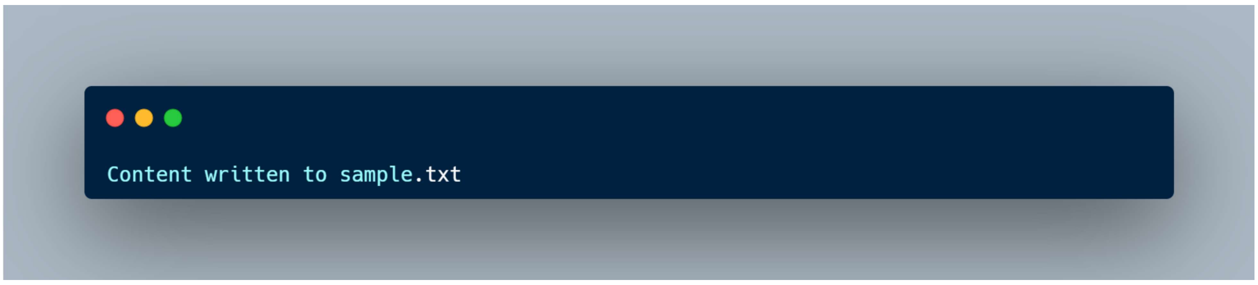
Code:



```python
with open("sample.txt", "w") as file:
    file.write("This is a sample text written to the file.\n")
    file.write("Python file handling is easy!\n")

print("Content written to sample.txt")
```

Output:



```
Content written to sample.txt
```

Explanation:

- **open("sample.txt", "w"):**
  - Opens a file named **sample.txt** in **write mode ("w")**.
  - If the file doesn't exist, it is created.
  - If the file exists, its contents are **erased**.
- **with statement:**
  - Ensures the file is automatically **closed** after the block is done (even if an error occurs).
  - This is safer and cleaner than manually calling file.close().
- **file** is the file object used to write content.

- Writes the first line of text into the file, ending with a newline (\n).
- Writes a second line to the file.
- Outputs a confirmation message after writing is done.

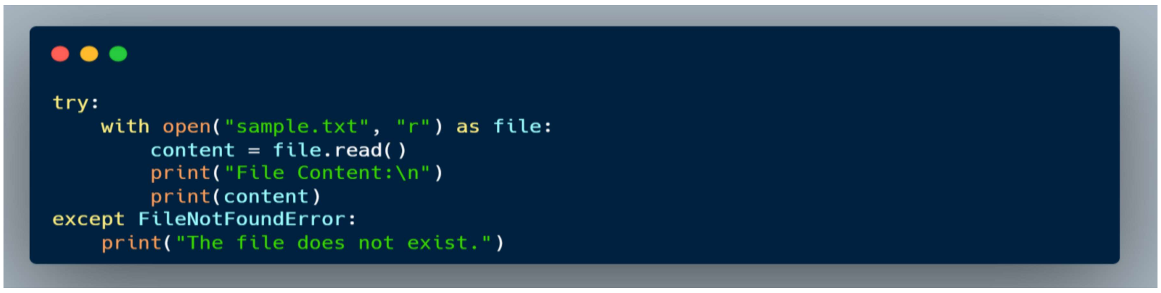
What Happens After Running the Code?

- A file named **sample.txt** has been created.
- Its content will be:

4. Read from a File

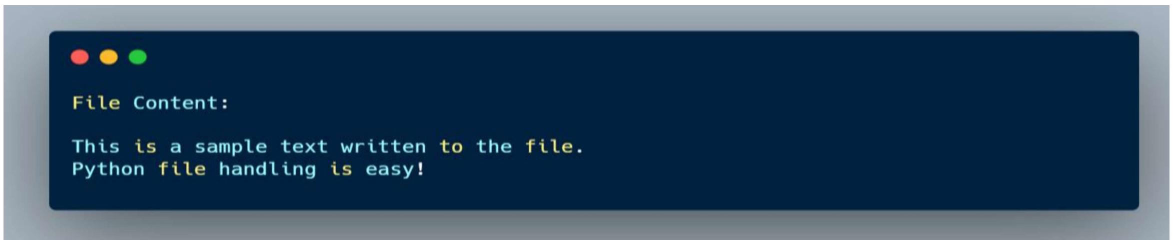
We used open in read mode and file.read to read and print to display.

Code:



```
try:
    with open("sample.txt", "r") as file:
        content = file.read()
        print("File Content:\n")
        print(content)
except FileNotFoundError:
    print("The file does not exist.")
```

Output:



```
File Content:

This is a sample text written to the file.
Python file handling is easy!
```

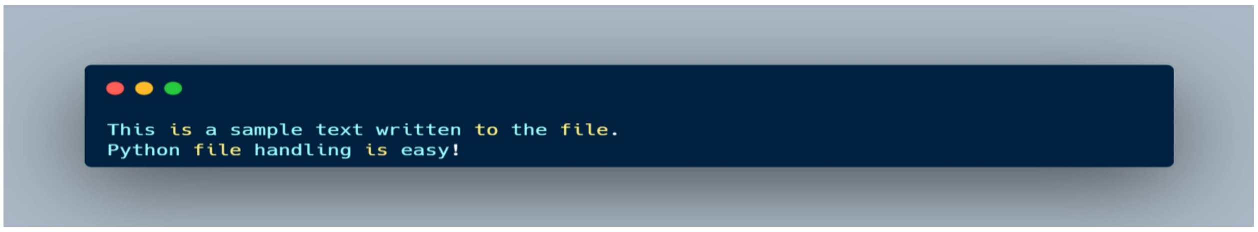
Explanation:

What the Code Does

- Tries to open and read a file named sample.txt
- Prints the file's content
- If the file doesn't exist, it handles the error gracefully
  - o **try:** starts a block of code that might cause an error.
  - o **open("sample.txt", "r")** opens the file in **read mode** ("r").
  - o If the file exists, it will proceed.
  - o If not, a FileNotFoundError is raised.
  - o **file.read()** reads the **entire content** of the file into the variable content.
  - o **print(content)** prints the content to the screen.
- Catches the case when the file does **not exist**, and shows a user-friendly message instead of crashing.

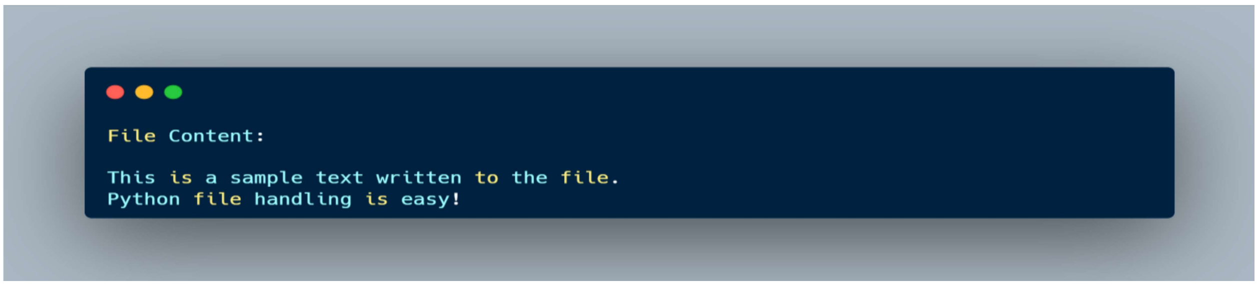
Example Output (if file exists)

If sample.txt contains:



```
This is a sample text written to the file.
Python file handling is easy!
```

Then the output will be:

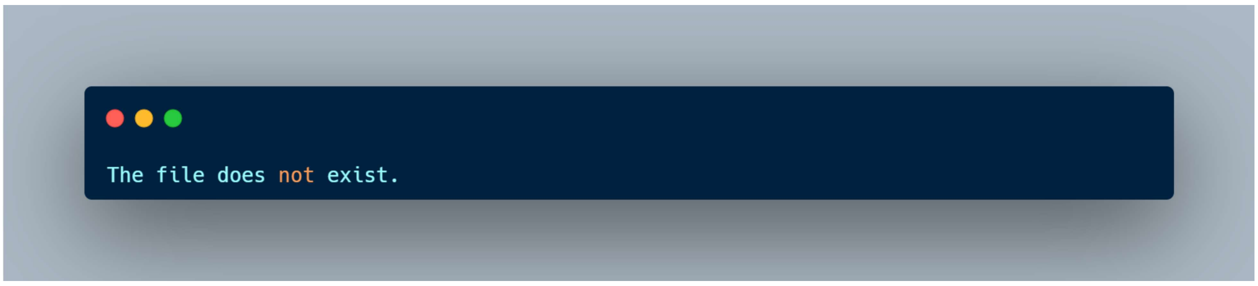


```
File Content:

This is a sample text written to the file.
Python file handling is easy!
```

If File is Missing

If sample.txt doesn't exist:



```
The file does not exist.
```