

Title: Python Execution Mode File Mode

Description: File Mode in Python is used to write and execute a complete Python program in a file with the .py extension. File Mode allows us to write multiple lines of code, save them, and run the entire program together. It is useful for creating programs that can be saved and used again later.

Purpose: The purpose of this practical is to learn how to create, write, save, and execute a Python program using File Mode. It helps students understand how Python programs are stored in files and how a complete program can be executed whenever required.

File Based Scripting Mode

Scenarios:

Practical 1 : Spilt the Restaurant Bill

Objective : To divide a total restaurant bill equally among a given number of friends.

```
total_bill = float(input("Enter total bill amount: "))
friends = int(input("How many friends? "))
per_person = total_bill / friends
print("Each person has to pay: Rs.", per_person)
```

Output:

```
Enter total bill amount: 1000
How many friends? 4
Each person has to pay: Rs. 250.0
```

Code Explanation :

- `total_bill = float(input("Enter total bill amount: "))`
 - This line asks the user to type in the total bill. The `input()` command displays the message on the screen. The `float()`

command takes whatever the user types and turns it into a number that can have decimals (like 1050.50). Finally, it saves this number in a container (called a variable) named `total_bill`.

- `friends = int(input("How many friends? "))`
 - This line asks the user how many people are splitting the bill. The `int()` command makes sure the answer is saved as a whole number (since you can't have half a person!). It saves this number in a container named `friends`.
- `per_person = total_bill / friends`
 - This is where the math happens. The computer takes the `total_bill` amount and divides it (using the `/` symbol) by the number of `friends`. It saves the calculated result in a new container called `per_person`.
- `print("Each person has to pay: Rs.", per_person)`
 - This line displays the final answer on the screen. It first prints the exact text "Each person has to pay: Rs.", and then adds the calculated amount from the `per_person` container right next to it.

Practical 2 : Simple ID Card Printer

Objective: To collect user data and print it in a structured, visual format resembling an ID card.

```
name = input("Enter name: ")
roll = input("Enter roll number: ")
school = input("Enter school name: ")
print("--- ID CARD ---")
print("Name   :", name)
print("Roll No:", roll)
print("School :", school)
print("-----")
```

Output:

```
Enter name: John Doe
Enter roll number: 42
Enter school name: Global High School
--- ID CARD ---
Name   : John Doe
Roll No: 42
School : Global High School
-----
```

Code Explanation:

1. `name = input("Enter name: ")` This line displays the message "Enter name: " to the user. It waits for the user to type something and press Enter. Whatever the user types is then saved into a variable called `name`.
2. `roll = input("Enter roll number: ")` Similar to the first line, this prompts the user with the message "Enter roll number: ". The user's typed response is stored in a variable called `roll`. (*Note: The `input()` function always captures data as a string, even if the user types numbers*).
3. `school = input("Enter school name: ")` This prompts the user with "Enter school name: " and stores the entered text into a variable called `school`.
4. `print("--- ID CARD ---")` This simply prints the literal text --- ID CARD -- to the screen, serving as a decorative header for the output.
5. `print("Name :", name)` This prints the text "Name :" followed by a space, and then the actual value that was stored in the `name` variable in step 1.
6. `print("Roll No:", roll)` This prints the text "Roll No:" followed by a space, and then the value stored in the `roll` variable in step 2.
7. `print("School :", school)` This prints the text "School :" followed by a space, and then the value stored in the `school` variable in step 3.
8. `print("-----")` This prints a series of dashes to create a bottom border, visually completing the ID card.

Practical 3 : Calculate Age from Birth Year

Objective: To calculate a person's current age by subtracting their birth year from the current year.

```
current_year = int(input("Enter current year: "))
birth_year = int(input("Enter birth year: "))
age = current_year - birth_year
print("You are", age, "years old.")
```

Output:

```
Enter current year: 2026
Enter birth year: 2000
You are 26 years old.
```

Code Explanation:

1. `current_year = int(input("Enter current year: "))`
 - `input("Enter current year: ")`: This prompts the user with the message "Enter current year: " and waits for them to type a response. By default, `input()` captures the user's response as a text string.
 - `int(...)`: Because we cannot perform mathematical subtraction on text, the `int()` function wraps around the input to convert the user's typed string into an integer (a whole number).
 - `current_year =`: The converted number is then stored in the variable named `current_year`.
2. `birth_year = int(input("Enter birth year: "))`
 - Similar to the first line, this prompts the user for their birth year, converts their typed text into an integer, and stores it in the variable `birth_year`.
3. `age = current_year - birth_year`

- This performs the mathematical calculation. It subtracts the integer stored in `birth_year` from the integer stored in `current_year`. The resulting number is stored in a new variable called `age`.

4. `print("You are", age, "years old.")`

- This line outputs the final result to the screen.
- The `print()` function is passed three separate arguments separated by commas: the string "You are", the calculated variable `age`, and the string "years old.". Python automatically inserts spaces between these arguments when displaying them.

Practical 4: Add Three Numbers

Objective: To take three separate numbers as input from the user and calculate their total sum.

```
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
c = int(input("Enter third number: "))
print("Total sum is:", a + b + c)
```

Output:

```
Enter first number: 10
Enter second number: 20
Enter third number: 30
Total sum is: 60
```

Code Explanation:

1. `a = int(input("Enter first number: "))`
 - `input("Enter first number: ")`: This prompts the user with the message "Enter first number: " and captures their response as text.
 - `int(...)`: This converts the user's text input into an integer (a whole number) so that math can be performed on it.
 - `a =`: The resulting integer is stored in the variable named `a`.

2. `b = int(input("Enter second number: "))`
 - Following the exact same logic, this line asks the user for a second number, converts it to an integer, and stores it in the variable `b`.
3. `c = int(input("Enter third number: "))`
 - This line asks for a third number, converts it, and stores it in the variable `c`.
4. `print("Total sum is:", a + b + c)`
 - This line handles both the final calculation and the output.
 - `a + b + c`: Python calculates the sum of the three integers stored in variables `a`, `b`, and `c`.
 - The `print()` function then displays the text "Total sum is:" followed immediately by the calculated sum.

Practical 5 : Calculate Result of student

Objective: To calculate the total marks and percentage for three subjects.

```
m1 = float(input("Subject 1 marks: "))
m2 = float(input("Subject 2 marks: "))
m3 = float(input("Subject 3 marks: "))

total = m1 + m2 + m3
percentage = (total / 300) * 100

print("Total Marks:", total)
print("Percentage:", percentage, "%")
```

Output:

```
Subject 1 marks: 85.5
Subject 2 marks: 90.0
Subject 3 marks: 94.5
Total Marks: 270.0
Percentage: 90.0 %
```

Code Explanation:

1. `m1 = float(input("Subject 1 marks: "))`
 - `input("Subject 1 marks: ")`: Prompts the user to enter their marks for the first subject.
 - `float(...)`: Converts the user's text input into a "float" (a floating-point number), which allows for decimals (e.g., 85.5) rather than just whole numbers.
 - `m1 =`: Stores this decimal number in the variable `m1`.
2. `m2 = float(input("Subject 2 marks: "))`
 - Exactly like the first line, this asks for the second subject's marks, converts the input to a decimal number, and stores it in the variable `m2`.
3. `m3 = float(input("Subject 3 marks: "))`
 - Asks for the third subject's marks, converts to a decimal, and stores it in the variable `m3`.
4. `total = m1 + m2 + m3`
 - Adds the three float values stored in `m1`, `m2`, and `m3` together, and stores the resulting sum in a new variable called `total`.
5. `percentage = (total / 300) * 100`
 - Calculates the average/percentage by dividing the total score by 300 (assuming each of the 3 subjects has a maximum score of 100).
 - It then multiplies the result by 100 to get the percentage format and stores it in the variable `percentage`.
6. `print("Total Marks:", total)`
 - Outputs the text "Total Marks:" followed by the calculated sum stored in the `total` variable.
7. `print("Percentage:", percentage, "%")`
 - Outputs the text "Percentage:", followed by the calculated value in the `percentage` variable, and finally adds the "%" symbol at the end.

Note: Open Python IDLE to perform the practical. The steps for opening IDLE and executing the code are already provided in the book.