



TECHNIK NEST

INNOVATIVE MINDS, NESTING SUCCESS

Name: Umme Habiba

Intern ID: TN/IN01/PY004

Email ID : saeedhabiba001@gmail.com

Internship Domain : Python Internee

Task Week : 2nd

Instructor Name : Hassan Ali

Task 1 :

Create a mini profile for a fictional user using variables. Store the following information:

Full name , Age , Current year, Country, Hobby, Expected graduation year (calculate it from current year + 4)

Print all details in a proper sentence format.

Also print how many years are left till graduation.

Solution :

Code Snippet & Screenshot

```
#Declaring variables
#mini profile for fictional user
#i want to be fictional character for my code
name = "habiba"
age = 21
current_year = 2025
hobby = "coding"
country = "Pakistan"
expected_graduation_year = current_year + 4
years_left = expected_graduation_year - current_year
print("i am", name)
print("i am ", age , "years old")
print("i live in", country)
print("i like", hobby, "in my free time")
print("inshAllah my degree is expected to be completed in",
expected_graduation_year , "years")
print("years left to my degree", years_left)
```

The screenshot displays the Visual Studio Code interface. The Explorer panel on the left shows the file structure. The main editor window contains a Python file named 'Task_1 Declaring Variables.py' with the following code:

```
1 #Declaring variables
2 #mini profile for fictional user
3 #i want to be fictional character for my code
4 name = "habiba"
5 age = 21
6 current_year = 2025
7 hobby = "coding"
8 country = "Pakistan"
9 expected_graduation_year = current_year + 4
10 years_left = expected_graduation_year - current_year
11
12 print("i am", name)
13 print("i am ", age ,"years old")
14 print("i live in", country)
15 print("i like", hobby, "in my free time")
```

The TERMINAL panel at the bottom shows the output of the script:

```
PS C:\Users\ff116cs027.pafast> & "C:/Program Files/Python313/python.exe" "C:/Users/ff116cs027.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositories/Python_Internship_Week2_Tasks/Task_1 Declaring Variables.py"
i am habiba
i am 21 years old
i live in Pakistan
i like coding in my free time
inshAllah my degree is expected to be completed in 2029 years
years left to my degree 4
PS C:\Users\ff116cs027.pafast> & "C:/Program Files/Python313/python.exe" "C:/Users/ff116cs027.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositories/Python_Internship_Week2_Tasks/Task_1 Declaring Variables.py"
i am habiba
i am 21 years old
```

Explanation:

- First I created mini profile using different characteristics.
- Along with that I **calculated** to estimate the graduation year by **adding 4** to the current year as given instructions.
- Then I calculated the **difference between the graduation year and current year**, which gives the number of years **left till graduation**
- Print function prints all information in a paragraph.

Task 2 :

Create 3 different user profiles (using variables). For each profile, include:

Name, profession, country, is_employed (Boolean)

Print their data in a tabular format using print() (not with external libraries).

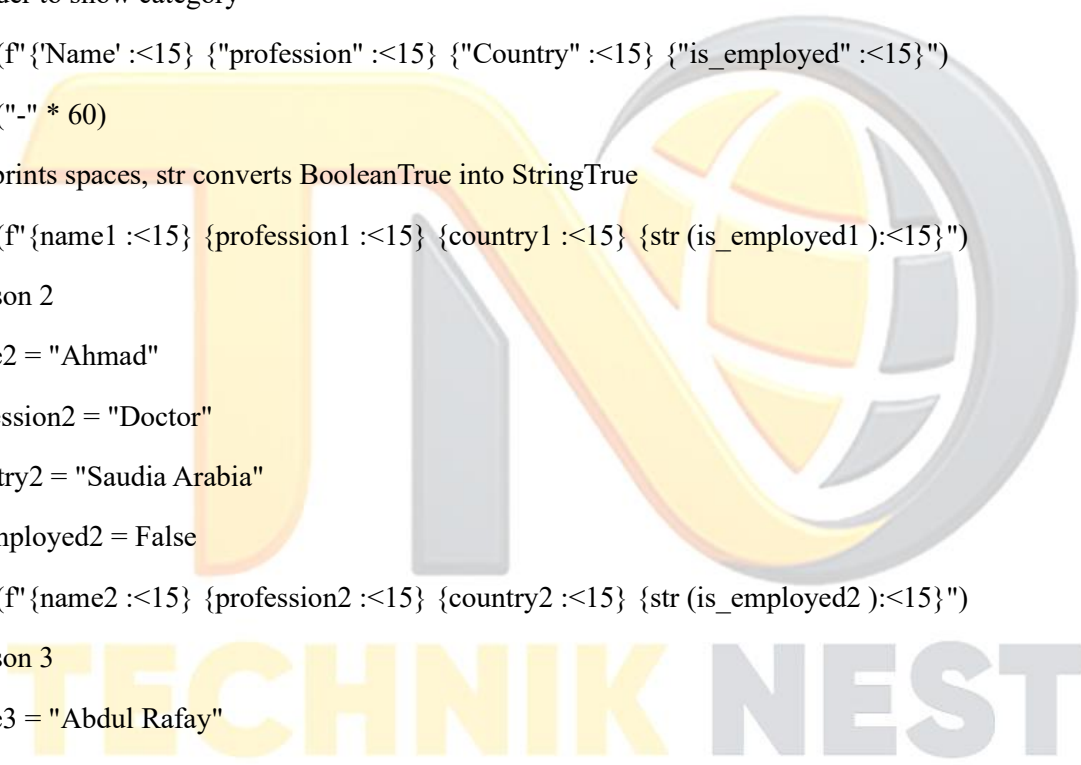
Solution :

Code Snippet & Screenshot

```
#let's create porfolio

#person 1
name1 = "Muhammad"
profession1 = "Engineer"
country1 = "Pakistan"
is_employed1 = True

#header to show category
print(f'{'Name' :<15} {"profession" :<15} {"Country" :<15} {"is_employed" :<15}')
```



```
print("-" * 60)
#15 prints spaces, str converts BooleanTrue into StringTrue
print(f'{'name1' :<15} {'profession1' :<15} {'country1' :<15} {'str (is_employed1) :<15}')
```

```
#person 2
name2 = "Ahmad"
profession2 = "Doctor"
country2 = "Saudia Arabia"
is_employed2 = False
print(f'{'name2' :<15} {'profession2' :<15} {'country2' :<15} {'str (is_employed2) :<15}')
```

```
#person 3
name3 = "Abdul Rafay"
profession3 = "Scientist"
country3 = "China"
is_employed3 = True
print(f'{'name3' :<15} {'profession3' :<15} {'country3' :<15} {'str (is_employed3) :<15}')
```

The image displays two screenshots of a Visual Studio Code editor window. The top screenshot shows a Python script with a syntax error. The bottom screenshot shows the same script after correction, with the output displayed in the terminal.

Top Screenshot (Syntax Error):

```
15 #person 2
16 name2 = "Ahmad"
17 profession2 = "Doctor"
18 country2 = "Saudia Arabia"
19 is_employed2 = False
20 print(f"{name2 :<15} {profession2 :<15} {country2 :<15} {str(is_employed2) :<15}")
21
22 #person 3
23 name3 = "Abdul Rafay"
24 profession3 = "Scientist"
25 country3 = "China"
26 is_employed3 = True
27 print(f"{name3 :<15} {profession3 :<15} {country3 :<15} {str(is_employed3) :<15}")
```

Bottom Screenshot (Corrected Script and Output):

```
15 #person 2
16 name2 = "Ahmad"
17 profession2 = "Doctor"
18 country2 = "Saudia Arabia"
19 is_employed2 = False
20 print(f"{name2 :<15} {profession2 :<15} {country2 :<15} {str(is_employed2) :<15}")
21
22 #person 3
23 name3 = "Abdul Rafay"
24 profession3 = "Scientist"
25 country3 = "China"
26 is_employed3 = True
27 print(f"{name3 :<15} {profession3 :<15} {country3 :<15} {str(is_employed3) :<15}")
```

Terminal Output:

```
Variables.py, line 9
print(f"{name2 :<15} {profession2 :<15} {country2 :<15} {str(is_employed2) :<15}")

SyntaxError: invalid syntax
PS C:\Users\fi116cs027\pafast> & "C:/Program Files/Python313/python.exe" "C:/Users/fi116cs027\pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositories/Python_Internship_Week2_Tasks/Task 2 Declaring Variables.py"
Name      profession  Country    is_employed
-----
Muhammad  Engineer   Pakistan   True
Ahmad     Doctor     Saudia Arabia False
Abdul Rafay Scientist   China      True
```

Explanation:

- Created profiles for each person.
- Create Header for table using simple Print() function.
- Print all profiles in tabular form using “f” string and table spaces.

Task 3 :

Declares five different variables , Stores a different data type in each (e.g., string, integer, float, boolean, complex)

Prints their values and data types

Then, converts each variable to a different type (where possible) and prints the new types

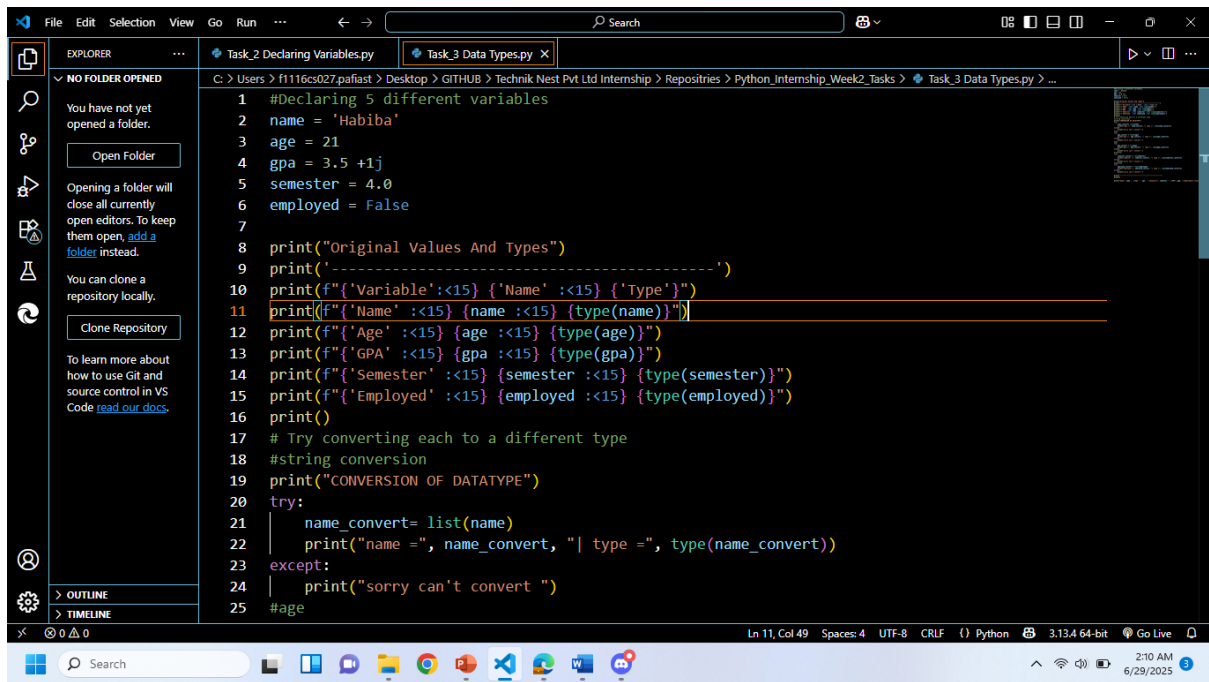
Note: You may not be able to convert all types — handle errors or comment why.

Solution :

Code Snippet & Screenshot

```
#Declaring 5 different variables
name = 'Habiba'
age = 21
gpa = 3.5 + 1j
semester = 4.0
employed = False
print("Original Values And Types")
print('-----')
print(f'{'Variable':<15} {'Name':<15} {'Type':<15}')
print(f'{'Name':<15} {name:<15} {type(name):<15}')
print(f'{'Age':<15} {age:<15} {type(age):<15}')
print(f'{'GPA':<15} {gpa:<15} {type(gpa):<15}')
print(f'{'Semester':<15} {semester:<15} {type(semester):<15}')
print(f'{'Employed':<15} {employed:<15} {type(employed):<15}')
print()
# Try converting each to a different type
#string conversion
print("CONVERSION OF DATATYPE")
try:
    name_convert= list(name)
    print("name =", name_convert, "| type =", type(name_convert))
except:
    print("sorry can't convert ")
#age
try:
    age_convert = float(age)
```

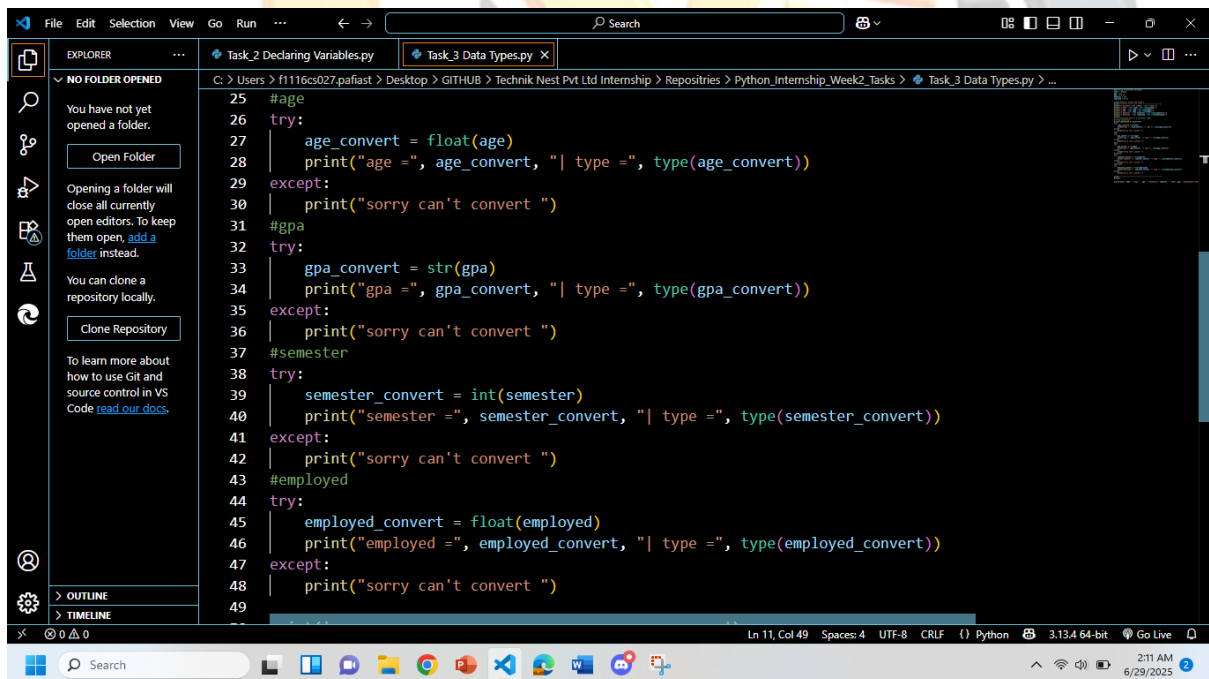
```
    print("age =", age_convert, "| type =", type(age_convert))
except:
    print("sorry can't convert ")
#gpa
try:
    gpa_convert = str(gpa)
    print("gpa =", gpa_convert, "| type =", type(gpa_convert))
except:
    print("sorry can't convert ")
#semester
try:
    semester_convert = int(semester)
    print("semester =", semester_convert, "| type =", type(semester_convert))
except:
    print("sorry can't convert ")
#employed
try:
    employed_convert = float(employed)
    print("employed =", employed_convert, "| type =", type(employed_convert))
except:
    print("sorry can't convert ")
print('-----')
print()
print("Name", name , "\nAge " , age , "\nSemester", semester, " \nGPA", gpa,
"\nEmployment status ", employed)
```

The screenshot shows the VS Code editor with two tabs: 'Task_2 Declaring Variables.py' and 'Task_3 Data Types.py'. The active tab is 'Task_3 Data Types.py', which contains the following code:

```
1 #Declaring 5 different variables
2 name = 'Habiba'
3 age = 21
4 gpa = 3.5 +1j
5 semester = 4.0
6 employed = False
7
8 print("Original Values And Types")
9 print('-----')
10 print(f'{"Variable":<15} {"Name":<15} {"Type":<15}')
11 print(f'{"Name":<15} {name:<15} {type(name)}')
12 print(f'{"Age":<15} {age:<15} {type(age)}')
13 print(f'{"GPA":<15} {gpa:<15} {type(gpa)}')
14 print(f'{"Semester":<15} {semester:<15} {type(semester)}')
15 print(f'{"Employed":<15} {employed:<15} {type(employed)}')
16 print()
17 # Try converting each to a different type
18 #string conversion
19 print("CONVERSION OF DATATYPE")
20 try:
21     name_convert= list(name)
22     print("name =", name_convert, "| type =", type(name_convert))
23 except:
24     print("sorry can't convert ")
25 #age
```

The status bar at the bottom indicates 'Ln 11, Col 49', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', '3.13.4 64-bit', and 'Go Live'.



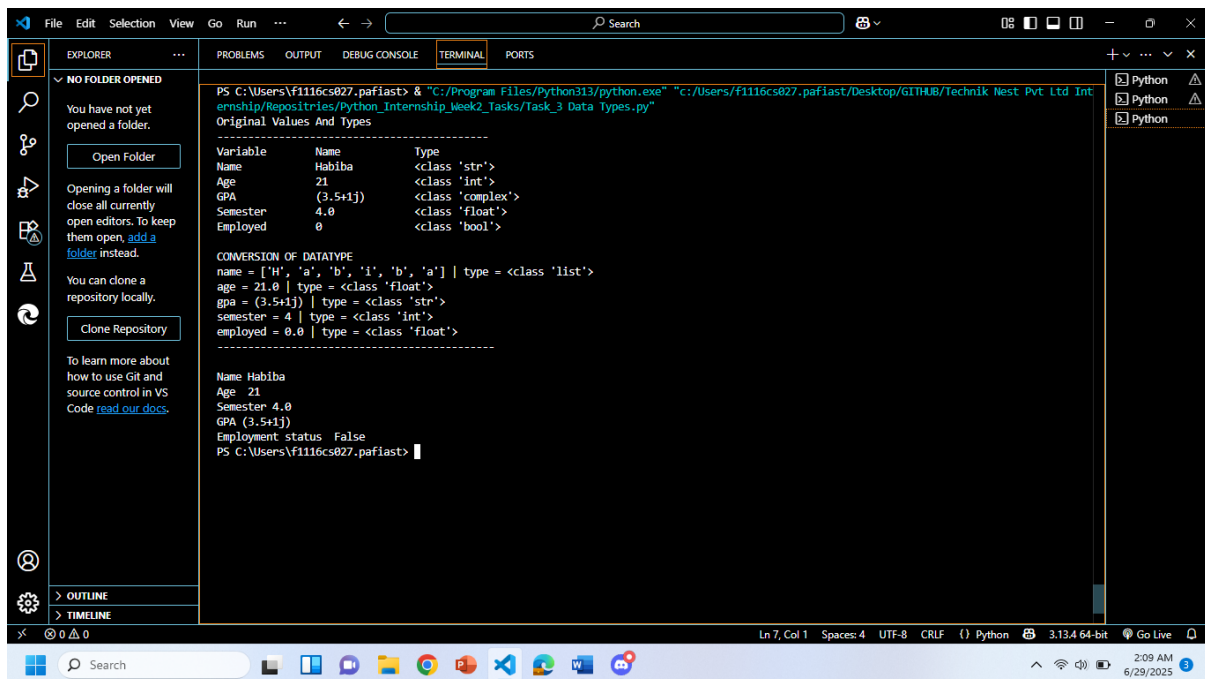
The screenshot shows the VS Code editor with the same two tabs. The active tab is 'Task_3 Data Types.py', which now shows the continuation of the code from the previous screenshot:

```
25 #age
26 try:
27     age_convert = float(age)
28     print("age =", age_convert, "| type =", type(age_convert))
29 except:
30     print("sorry can't convert ")
31 #gpa
32 try:
33     gpa_convert = str(gpa)
34     print("gpa =", gpa_convert, "| type =", type(gpa_convert))
35 except:
36     print("sorry can't convert ")
37 #semester
38 try:
39     semester_convert = int(semester)
40     print("semester =", semester_convert, "| type =", type(semester_convert))
41 except:
42     print("sorry can't convert ")
43 #employed
44 try:
45     employed_convert = float(employed)
46     print("employed =", employed_convert, "| type =", type(employed_convert))
47 except:
48     print("sorry can't convert ")
49
```

The status bar at the bottom indicates 'Ln 11, Col 49', 'Spaces: 4', 'UTF-8', 'CRLF', 'Python', '3.13.4 64-bit', and 'Go Live'.

```
50 print('-----')
51 print()
52
53 print("Name", name , "\nAge " , age , "\nSemester", semester, " \nGPA", gpa, "\nEmployment statu
54
55
```


Output:



```
PS C:\Users\F1116cs027.pafiast> & "C:/Program Files/Python313/python.exe" "c:/Users/f1116cs027.pafiast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositories/Python_Internship_Week2_Tasks/Task_3 Data Types.py"
Original Values And Types
-----
Variable      Name      Type
-----
Name          Habiba    <class 'str'>
Age           21        <class 'int'>
GPA           (3.5+1j)  <class 'complex'>
Semester      4.0       <class 'float'>
Employed      0         <class 'bool'>

CONVERSION OF DATATYPE
name = ['h', 'a', 'b', 'i', 'b', 'a'] | type = <class 'list'>
age = 21.0 | type = <class 'float'>
gpa = (3.5+1j) | type = <class 'str'>
semester = 4 | type = <class 'int'>
employed = 0.0 | type = <class 'float'>

-----
Name Habiba
Age 21
Semester 4.0
GPA (3.5+1j)
Employment status False
PS C:\Users\F1116cs027.pafiast>
```

Explanation:

- First I declare 5 variables and then print their Original values and datatypes using Type.
- And then converted their types using Try & Except. Which will prevent the program from crash if some of them wont be able to convert their type like Complex.

Task 4 :

Create a data type tester:

Ask the user to input any value. , Detect and print what Python guesses its type as (use type()).

Add conditions to identify if it's likely an integer, float, or string, and print a message like:

"You entered a float!"

Solution :

Code Snippet & Screenshot

```
#create datatype tester

fan = input("enter any value ")

print("Python Guesses Type ", type(fan))
```

#let's add some condition to check different cases

try:

```
int(fan)
```

```
print("You entered an integer!")
```

except:

try:

```
float(fan)
```

```
print("you entered float")
```

except:

#try:

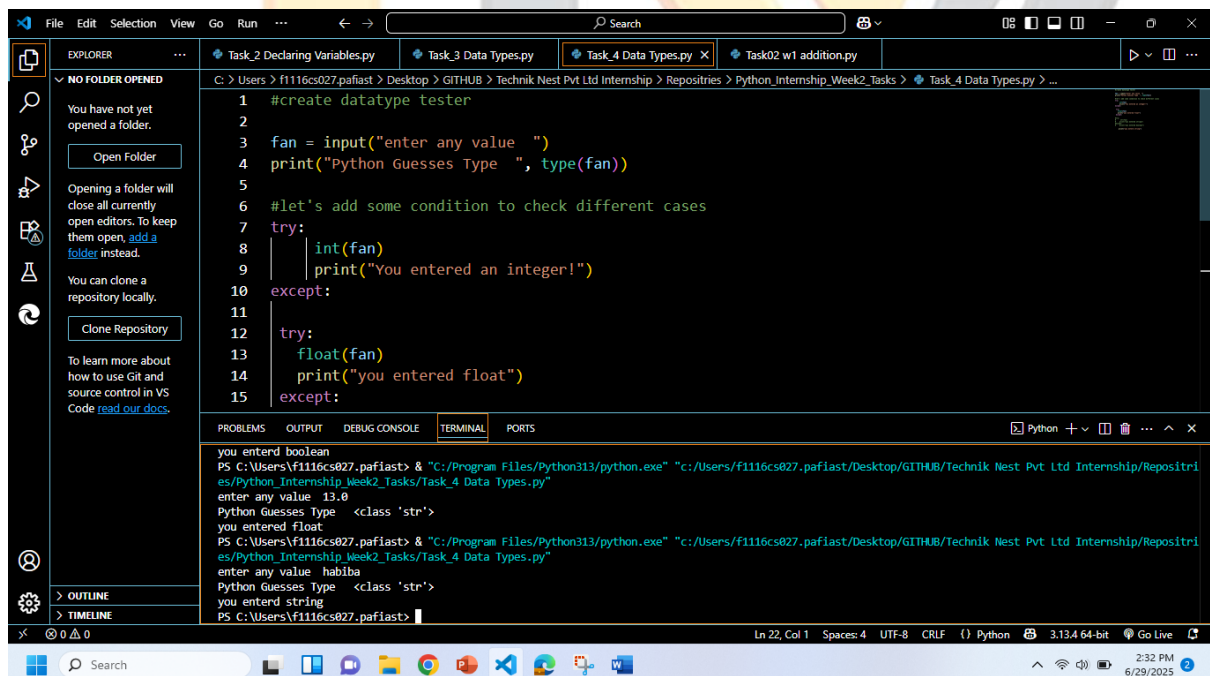
```
# str(fan)
```

```
# print("you entered string")
```

#except:

```
# print("you entered boolean")
```

```
print("you entered string")
```

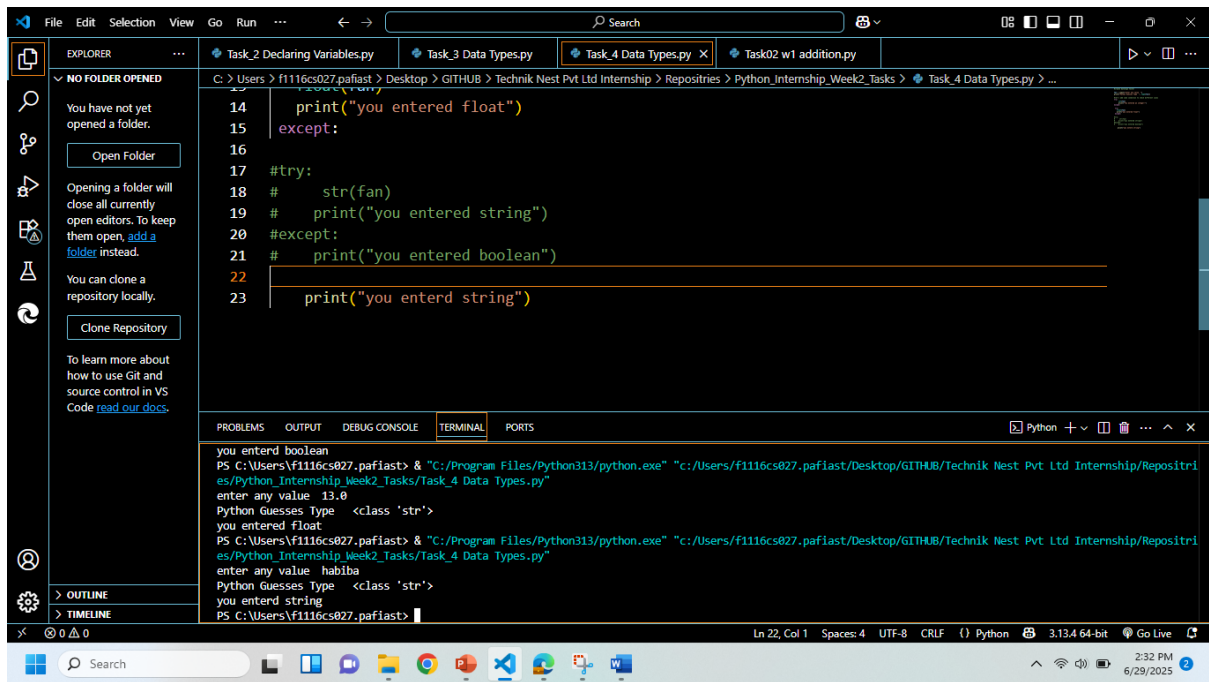


The screenshot shows a Visual Studio Code editor window with a Python file named 'Task_4 Data Types.py'. The code in the editor is as follows:

```
1 #create datatype tester
2
3 fan = input("enter any value ")
4 print("Python Guesses Type ", type(fan))
5
6 #let's add some condition to check different cases
7 try:
8     int(fan)
9     print("You entered an integer!")
10 except:
11
12     try:
13         float(fan)
14         print("you entered float")
15     except:
```

The terminal output shows the execution of the script. It prompts the user to enter a value, and the script correctly identifies the data type of the input.

```
you entered boolean
PS C:\Users\vf116cs827.pafast> & "C:/Program Files/Python313/python.exe" "c:/Users/vf116cs827.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositri
es/Python_Internship_Week2_Tasks/Task_4 Data Types.py"
enter any value 13.0
Python Guesses Type <class 'str'>
you entered float
PS C:\Users\vf116cs827.pafast> & "C:/Program Files/Python313/python.exe" "c:/Users/vf116cs827.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositri
es/Python_Internship_Week2_Tasks/Task_4 Data Types.py"
enter any value habiba
Python Guesses Type <class 'str'>
you entered string
PS C:\Users\vf116cs827.pafast>
```



Explanation:

- Create Tester, ask user to input value and let python guesses its data type.
- As python guesses every value as string Then I add Try & Except to check different datatypes when user enters anything.
- I did also use string in try but then I realize its already exist by default so I shift it to except.
- Now my code can recognize the 2 datatypes, int, float and off course string by default.

Task 5 :

Design a command-line survey that:

Asks the user 5 different questions (e.g., name, favorite food, birth year, favorite number, favorite language)

Then prints a summary of all responses in sentence format.

Use formatting to make the output look professional (e.g., f-strings).

Solution :

Code Snippet & Screenshot

```
#let's design command-line survey
```

```
#name, favorite food, birth year, favorite number, favorite language
```

```
name = input("1. May I know your full name? ")
```

```
print("oh that's great", name)
```

```

food = input("2. What is your favorite food? ")

print("Yum! That's a great choice", food)

birth_year = input("3. Kindly enter your birth year: ")

print("Interesting! Born in", birth_year )

fav_number = input("4. What is your favorite number? ")

print("Nice pick! Number", fav_number )

language = input("5. Which programming language do you prefer the most? ")

print(language,"is an excellent choice for modern development")

#summary

print("\nSURVEY SUMMARY")

print(f" {name} loves eating {food}.")

print(f" Born in {birth_year}, they hold a special liking for the number {fav_number}.")

print(f" When it comes to coding, {language} is their preferred language.")

print("=====")

print(" Thank you for participating in our survey!")

```

The screenshot shows a Visual Studio Code editor window with a Python file named 'Task_5 Input Output.py'. The code is a command-line survey script. The terminal at the bottom shows the execution of the script, with user input and program output.

```

1 #let's design command-line survey
2 #name, favorite food, birth year, favorite number, favorite language
3 name = input("1. May I know your full name? ")
4 print("oh that's great", name)
5 food = input("2. What is your favorite food? ")
6 print("Yum! That's a great choice", food)
7 birth_year = input("3. Kindly enter your birth year: ")
8 print("Interesting! Born in", birth_year )
9 fav_number = input("4. What is your favorite number? ")
10 print("Nice pick! Number", fav_number )
11 language = input("5. Which programming language do you prefer the most? ")
12 print(language,"is an excellent choice for modern development")
13
14 #summary
15 print("\nSURVEY SUMMARY")

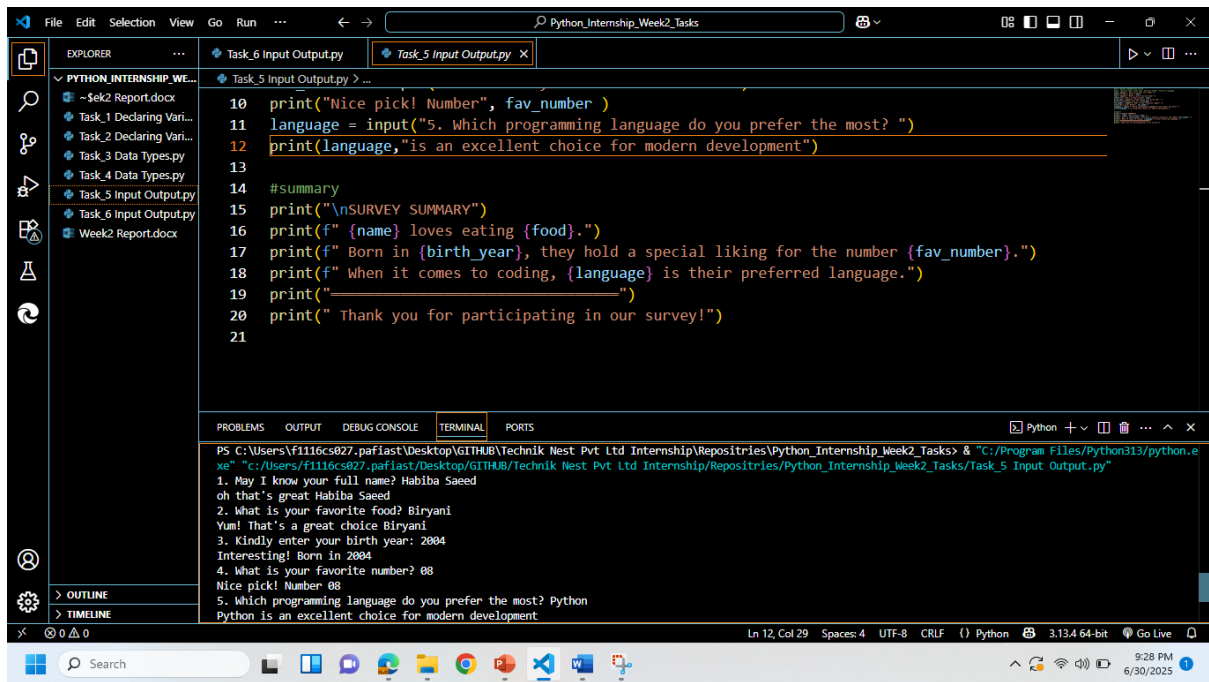
```

Terminal Output:

```

PS C:\Users\vf116cs827_paf\Ia\st\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/Python313/python.exe" "C:/Users/vf116cs827_paf\Ia\st\Desktop\GITHUB/Technik Nest Pvt Ltd Internship/Repositries/Python_Internship_Week2_Tasks/Task_5 Input Output.py"
1. May I know your full name? Habiba Saeed
oh that's great Habiba Saeed
2. What is your favorite food? Biryani
Yum! That's a great choice Biryani
3. Kindly enter your birth year: 2004
Interesting! Born in 2004
4. What is your favorite number? 08
Nice pick! Number 08
5. Which programming language do you prefer the most? Python
Python is an excellent choice for modern development

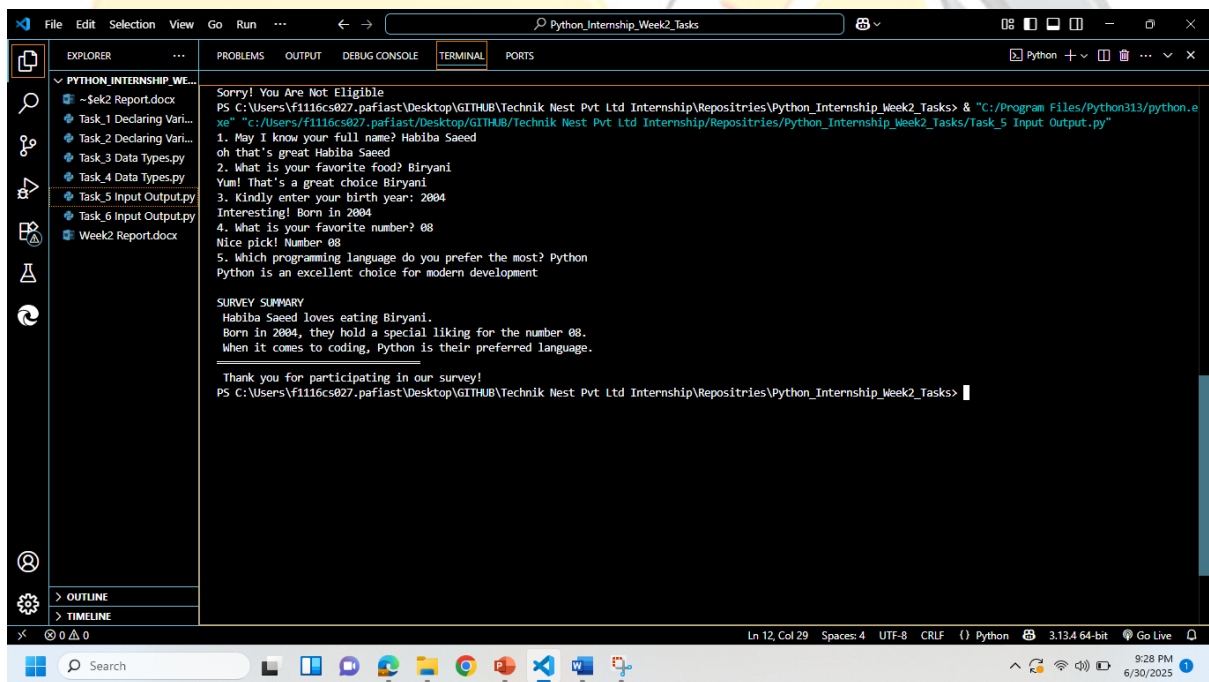
```



```
10 print("Nice pick! Number", fav_number )
11 language = input("5. Which programming language do you prefer the most? ")
12 print(language,"is an excellent choice for modern development")
13
14 #summary
15 print("\nSURVEY SUMMARY")
16 print(f" {name} loves eating {food}.")
17 print(f" Born in {birth_year}, they hold a special liking for the number {fav_number}.")
18 print(f" When it comes to coding, {language} is their preferred language.")
19 print("")
20 print(" Thank you for participating in our survey!")
21
```

PS C:\Users\F1116cs827.paf\ast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/Python313/python.exe" "C:/Users/F1116cs827.paf\ast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositries/Python_Internship_Week2_Tasks/Task_5 Input Output.py"

1. May I know your full name? Habiba Saeed
oh that's great Habiba Saeed
2. What is your favorite food? Biryani
Yum! That's a great choice Biryani
3. Kindly enter your birth year: 2004
Interesting! Born in 2004
4. What is your favorite number? 08
Nice pick! Number 08
5. Which programming language do you prefer the most? Python
Python is an excellent choice for modern development



Sorry! You Are Not Eligible

PS C:\Users\F1116cs827.paf\ast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/Python313/python.exe" "C:/Users/F1116cs827.paf\ast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositries/Python_Internship_Week2_Tasks/Task_5 Input Output.py"

1. May I know your full name? Habiba Saeed
oh that's great Habiba Saeed
2. What is your favorite food? Biryani
Yum! That's a great choice Biryani
3. Kindly enter your birth year: 2004
Interesting! Born in 2004
4. What is your favorite number? 08
Nice pick! Number 08
5. Which programming language do you prefer the most? Python
Python is an excellent choice for modern development

SURVEY SUMMARY
Habiba Saeed loves eating Biryani.
Born in 2004, they hold a special liking for the number 08.
When it comes to coding, Python is their preferred language.

Thank you for participating in our survey!

PS C:\Users\F1116cs827.paf\ast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> |

Explanation:

- I Take 5 questions from user as part of survey.
 - Then print it in paragraph in professional manner to show input / output.
-

Task 6 :

Ask the user to:

Enter their year of birth , Calculate their age (based on current year) ,Check if the user is eligible to vote (18+ years)

Display a message: "You are eligible to vote." or "You are not eligible to vote yet."

Solution :

Code Snippet & Screenshot

```
#eligibility for vote

#Enter their year of birth

print("Let's Check Your Eligibility For Vote")

birth= int(input("Enter Your Birth Year = "))

# Calculate their age (based on current year)

calculate = (2025- birth)

print(calculate)

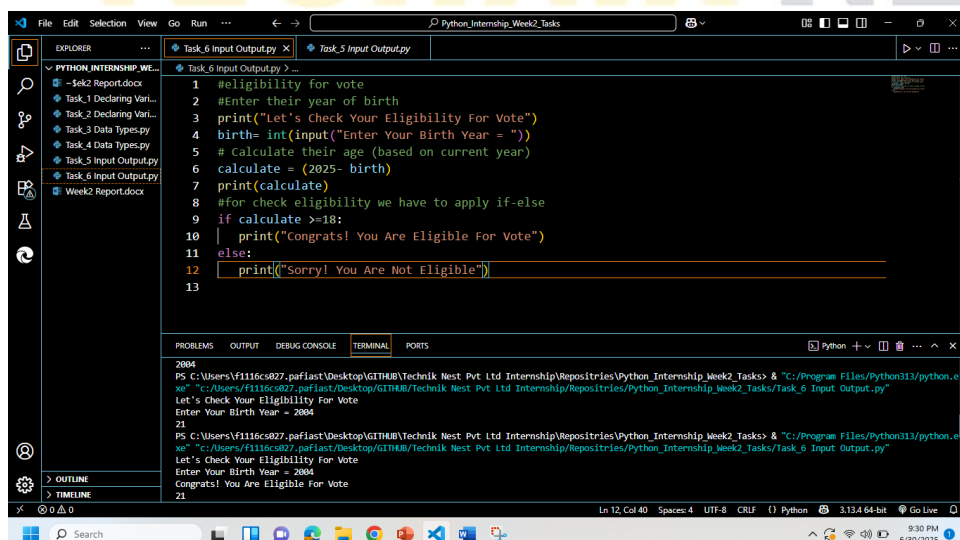
#for check eligibility we have to apply if-else

if calculate >=18:

    print("Congrats! You Are Eligible For Vote")

else:

    print("Sorry! You Are Not Eligible")
```



The screenshot shows a Python IDE with the following components:

- EXPLORER:** A file explorer on the left showing a project named 'PYTHON_INTERNSHIP_WEEK2'. It contains several files, including 'Task_6 Input Output.py' which is currently selected.
- EDITOR:** The main window displays the Python code for Task 6, which is identical to the code snippet provided above.
- TERMINAL:** The bottom panel shows the output of the program. It displays the prompt 'Let's Check Your Eligibility For Vote', the user input '2004', the calculated age '21', and the final message 'Congrats! You Are Eligible For Vote'.

Explanation:

- Take birth year from user.
 - Subtract it from current year to check the age of user.
 - Print their age.
 - And then displays a msg if user is eligible using If/Else statement.
-

Task 7:

Create a marks percentage calculator:

Ask user to input marks for 5 subjects (input as strings) , Convert them to integers

Calculate the total and percentage

Print percentage along with a grade: A (90+), B (80-89), C (70-79), Fail (<70)

Solution :

Code Snippet & Screenshot

```
#Create a marks percentage calculator:

#Ask user to input marks for 5 subjects (input as strings) , Convert them to integers
#let's use while loop to save time and make our code interesting one

total_marks = 0
count = 1
while count <=5:
    marks = float(input(f" Enter marks for subject {count}: "))
    total_marks += marks
    count+= 1

#Calculate the total and percentage
percentage = total_marks / 500 * 100

print(percentage)

#Print percentage along with a grade: A (90+), B (80-89), C (70-79), Fail (<70)
#we will use IF/ELSE statments

if percentage >=90:
    print("Grade A")

elif percentage >=80:
```



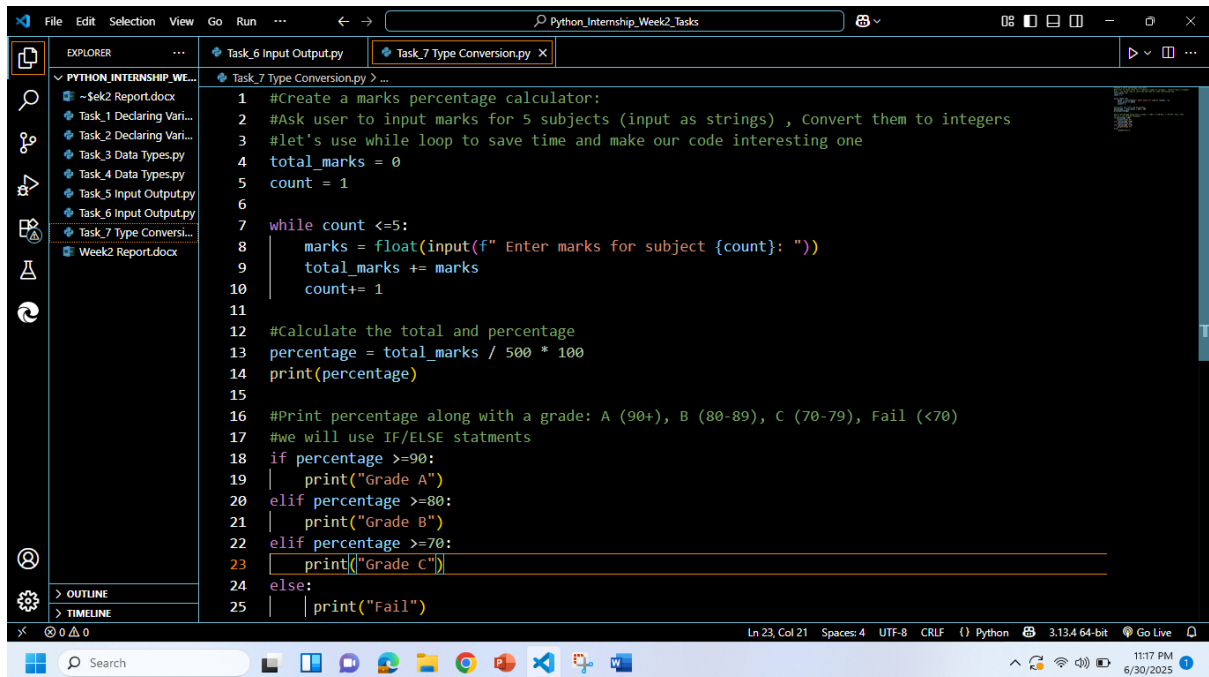
```
print("Grade B")

elif percentage >=70:

    print("Grade C")

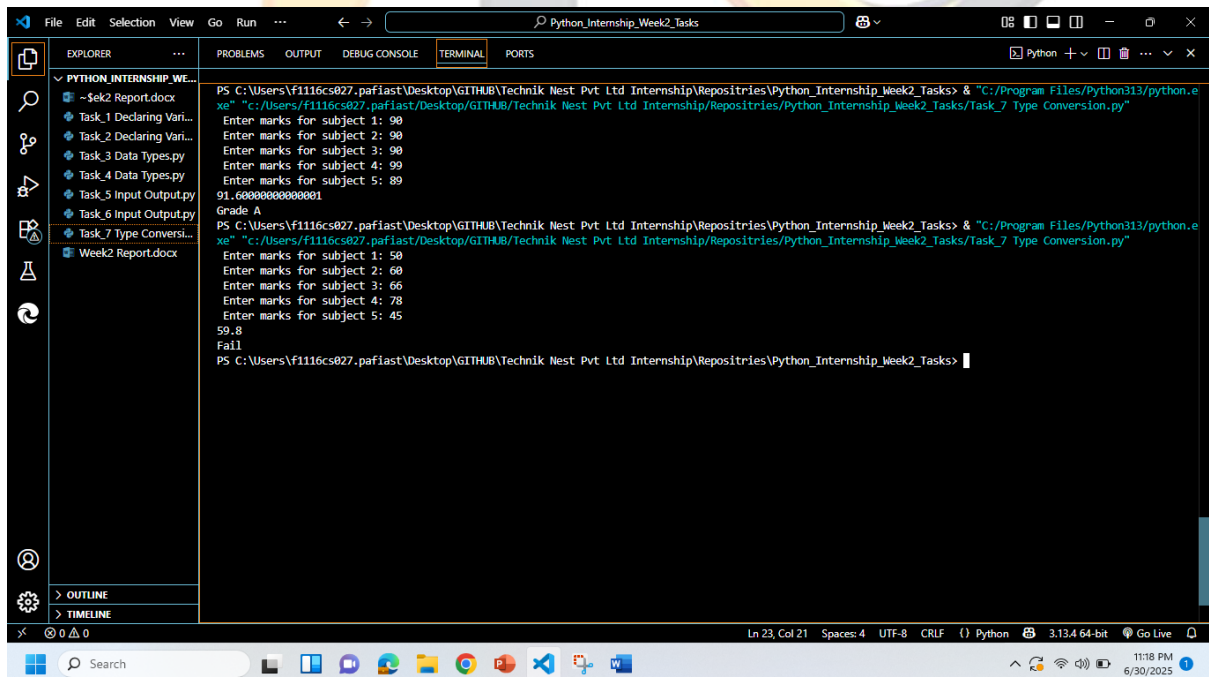
else:

    print("Fail")
```



The screenshot shows a code editor with the following Python code:

```
1 #Create a marks percentage calculator:
2 #Ask user to input marks for 5 subjects (input as strings) , Convert them to integers
3 #let's use while loop to save time and make our code interesting one
4 total_marks = 0
5 count = 1
6
7 while count <=5:
8     marks = float(input(f" Enter marks for subject {count}: "))
9     total_marks += marks
10    count+= 1
11
12 #Calculate the total and percentage
13 percentage = total_marks / 500 * 100
14 print(percentage)
15
16 #Print percentage along with a grade: A (90+), B (80-89), C (70-79), Fail (<70)
17 #we will use IF/ELSE statements
18 if percentage >=90:
19     print("Grade A")
20 elif percentage >=80:
21     print("Grade B")
22 elif percentage >=70:
23     print("Grade C")
24 else:
25     print("Fail")
```



The screenshot shows a terminal window with the following output:

```
PS C:\Users\F1116cs827.paf\ Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/Python313/python.exe" "C:/Users/F1116cs827.paf\ Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks/Task_7 Type Conversion.py"
Enter marks for subject 1: 90
Enter marks for subject 2: 90
Enter marks for subject 3: 90
Enter marks for subject 4: 99
Enter marks for subject 5: 89
91.60000000000001
Grade A
PS C:\Users\F1116cs827.paf\ Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/Python313/python.exe" "C:/Users/F1116cs827.paf\ Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks/Task_7 Type Conversion.py"
Enter marks for subject 1: 58
Enter marks for subject 2: 60
Enter marks for subject 3: 66
Enter marks for subject 4: 78
Enter marks for subject 5: 45
59.8
Fail
PS C:\Users\F1116cs827.paf\ Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks>
```

Explanation:

- I started from While Loop as it is quite boring to collect data in a similar way in every code.
- It also makes my code looks good and saves my time as well.
- I was stuck on count += 1 because that was not in the loop, and I kept on fixing other things.
- Then I counted percentage.
- Uses if/else to adjust the conditions for grading.

Task 8:

Create a temperature converter:

Ask the user to input temperature in Celsius.

Convert it to Fahrenheit using: $F = (C * 9/5) + 32$, Then reverse: Ask for Fahrenheit, convert it to Celsius.

Handle wrong input types using try-except.

Solution :

Code Snippet & Screenshot

```
#TEMPERATURE CONVERTER

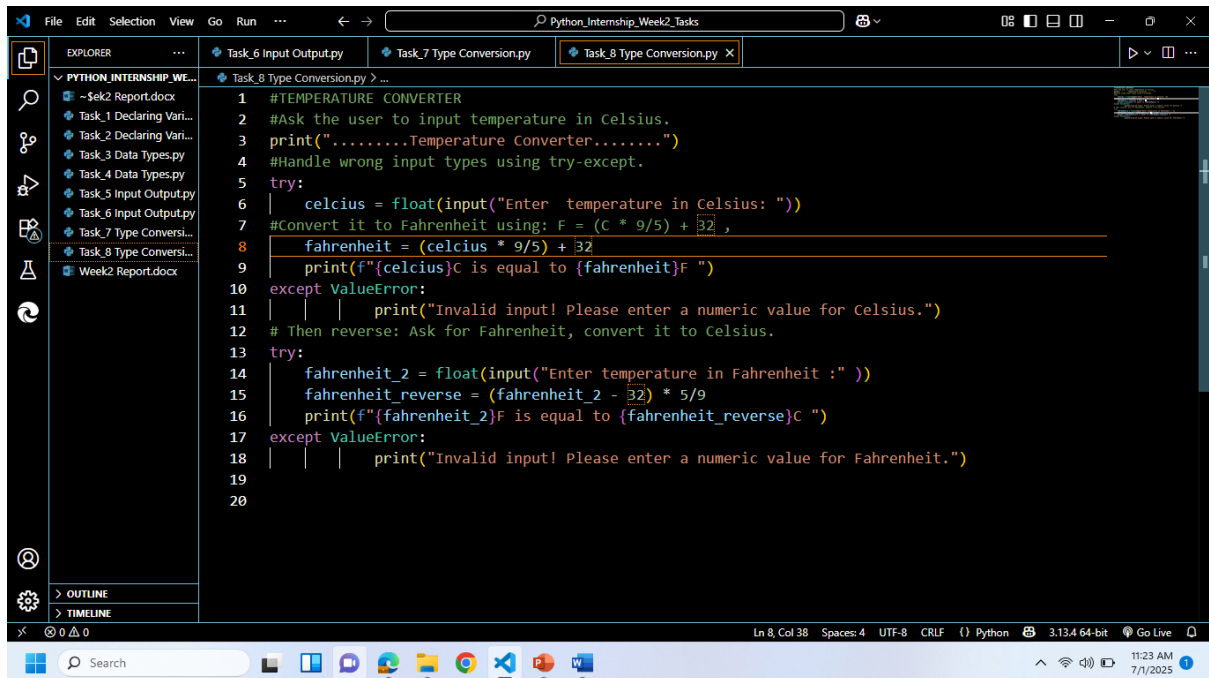
#Ask the user to input temperature in Celsius.
print(".....Temperature Converter.....")

#Handle wrong input types using try-except.
try:
    celcius = float(input("Enter temperature in Celsius: "))
#Convert it to Fahrenheit using:  $F = (C * 9/5) + 32$  ,
    fahrenheit = (celcius * 9/5) + 32
    print(f'{celcius}C is equal to {fahrenheit}F ')
except ValueError:
    print("Invalid input! Please enter a numeric value for Celsius.")

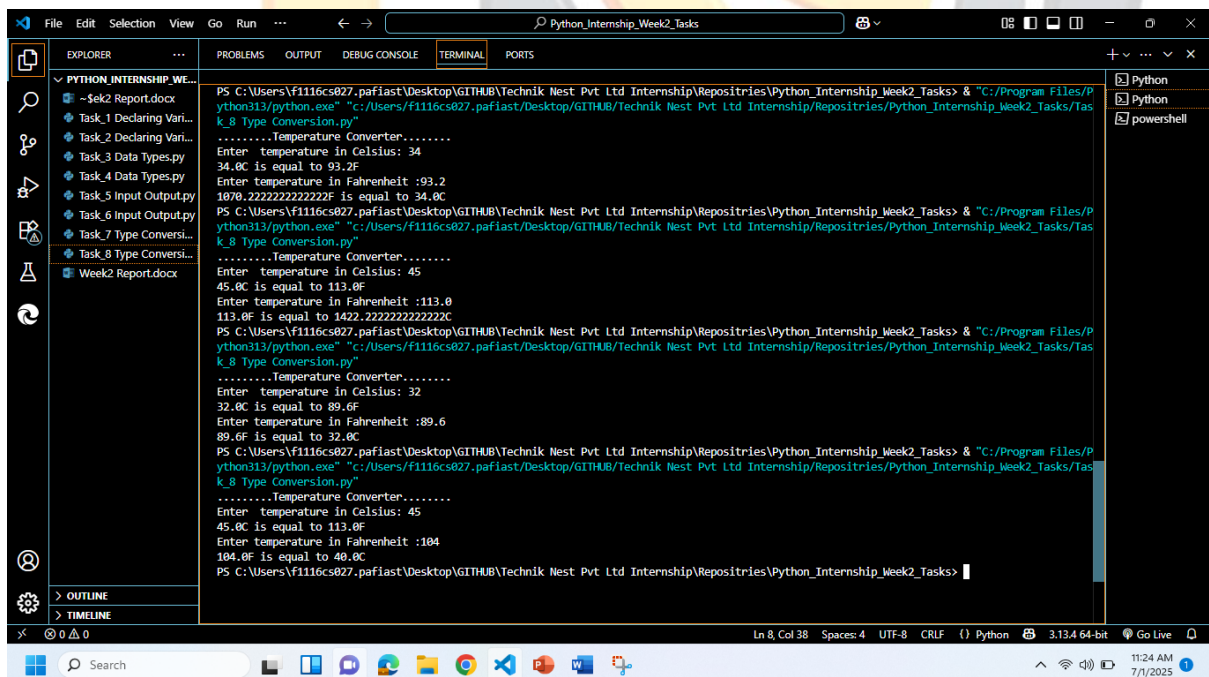
# Then reverse: Ask for Fahrenheit, convert it to Celsius.
try:
    fahrenheit_2 = float(input("Enter temperature in Fahrenheit :"))
    fahrenheit_reverse = (fahrenheit_2 - 32) * 5/9
    print(f'{fahrenheit_2}F is equal to {fahrenheit_reverse}C ")
```

except ValueError:

```
print("Invalid input! Please enter a numeric value for Fahrenheit.")
```



```
1 #TEMPERATURE CONVERTER
2 #Ask the user to input temperature in Celsius.
3 print(".....Temperature Converter.....")
4 #Handle wrong input types using try-except.
5 try:
6     celcius = float(input("Enter temperature in Celsius: "))
7     #Convert it to Fahrenheit using: F = (C * 9/5) + 32 ,
8     fahrenheit = (celcius * 9/5) + 32
9     print(f"{celcius}C is equal to {fahrenheit}F ")
10 except ValueError:
11     print("Invalid input! Please enter a numeric value for Celsius.")
12 # Then reverse: Ask for Fahrenheit, convert it to Celsius.
13 try:
14     fahrenheit_2 = float(input("Enter temperature in Fahrenheit :"))
15     fahrenheit_reverse = (fahrenheit_2 - 32) * 5/9
16     print(f"{fahrenheit_2}F is equal to {fahrenheit_reverse}C ")
17 except ValueError:
18     print("Invalid input! Please enter a numeric value for Fahrenheit.")
19
20
```



```
PS C:\Users\fi116cs827\pafast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/P
ython313/python.exe" "c:/Users/fi116cs827.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositries/Python_Internship_Week2_Tasks/Tas
k_8 Type Conversion.py"
.....Temperature Converter.....
Enter temperature in Celsius: 34
34.0C is equal to 93.2F
Enter temperature in Fahrenheit :93.2
109.02222222222222F is equal to 34.0C
PS C:\Users\fi116cs827.pafast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/P
ython313/python.exe" "c:/Users/fi116cs827.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositries/Python_Internship_Week2_Tasks/Tas
k_8 Type Conversion.py"
.....Temperature Converter.....
Enter temperature in Celsius: 45
45.0C is equal to 113.0F
Enter temperature in Fahrenheit :113.0
113.0F is equal to 142.2222222222222C
PS C:\Users\fi116cs827.pafast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/P
ython313/python.exe" "c:/Users/fi116cs827.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositries/Python_Internship_Week2_Tasks/Tas
k_8 Type Conversion.py"
.....Temperature Converter.....
Enter temperature in Celsius: 32
32.0C is equal to 89.6F
Enter temperature in Fahrenheit :89.6
89.6F is equal to 32.0C
PS C:\Users\fi116cs827.pafast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks> & "C:/Program Files/P
ython313/python.exe" "c:/Users/fi116cs827.pafast/Desktop/GITHUB/Technik Nest Pvt Ltd Internship/Repositries/Python_Internship_Week2_Tasks/Tas
k_8 Type Conversion.py"
.....Temperature Converter.....
Enter temperature in Celsius: 45
45.0C is equal to 113.0F
Enter temperature in Fahrenheit :104
104.0F is equal to 40.0C
PS C:\Users\fi116cs827.pafast\Desktop\GITHUB\Technik Nest Pvt Ltd Internship\Repositries\Python_Internship_Week2_Tasks>
```

Explanation:

- Take values from user and convert them using formulas.
- I did minor mistakes while calculations and run the terminal accordingly to fix errors.
- Try/except logic is also used for to handle wrong input.