



TECHNIK NEST

INNOVATIVE MINDS, NESTING SUCCESS

Name: Maaz Sher Muhammad

Intern ID: TN/1N01/003

Email ID : maazshermuhammadofficial@gmail.com

Internship Domain : Python Internee

Task Week : 4

Instructor Name : Hassan Ali

Task 1 :

Code Snippet & Screenshot

```
#ask user for input and check its type
user_input = input("Enter any value: ")
print("The data type of your input is:", type(user_input))
```

#execute a string as Python code

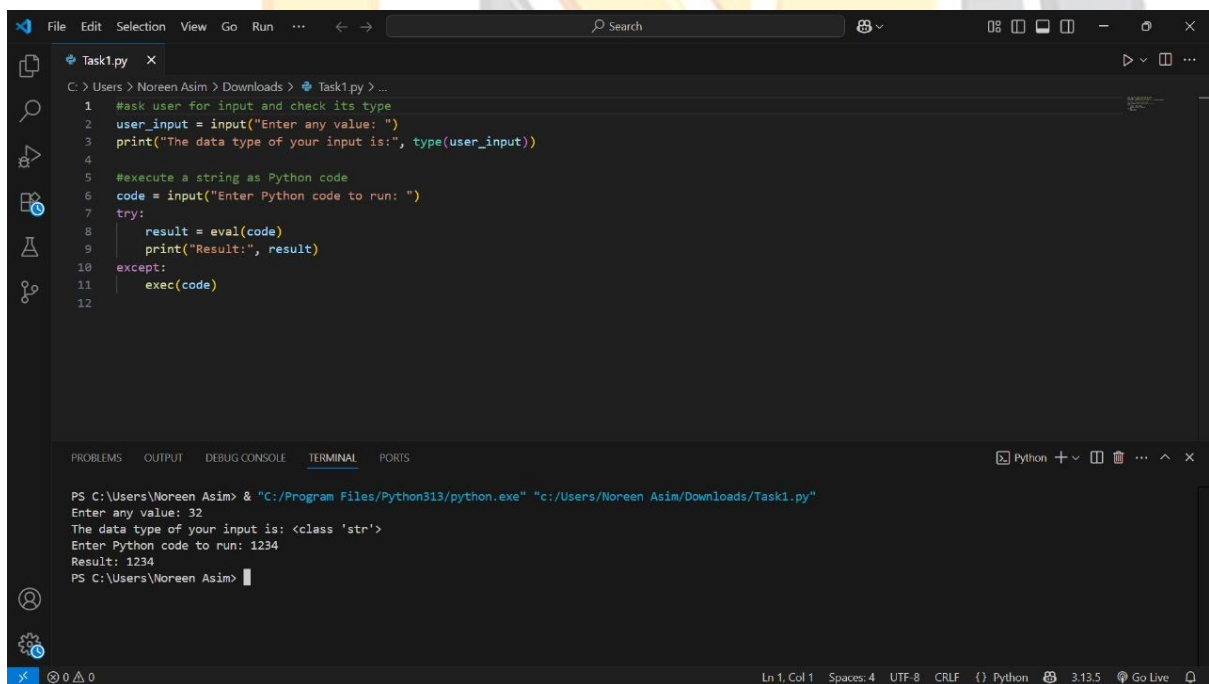
```
code = input("Enter Python code to run: ")
```

try:

```
    result = eval(code)
    print("Result:", result)
```

except:

```
    exec(code)
```



The screenshot shows a Visual Studio Code editor window with a file named 'Task1.py' open. The code in the editor is as follows:

```
1 #ask user for input and check its type
2 user_input = input("Enter any value: ")
3 print("The data type of your input is:", type(user_input))
4
5 #execute a string as Python code
6 code = input("Enter Python code to run: ")
7 try:
8     result = eval(code)
9     print("Result:", result)
10 except:
11     exec(code)
12
```

Below the editor, the 'TERMINAL' panel shows the execution of the script using Python 3.13.5. The output is as follows:

```
PS C:\Users\Noreen Asim> "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task1.py"
Enter any value: 32
The data type of your input is: <class 'str'>
Enter Python code to run: 1234
Result: 1234
PS C:\Users\Noreen Asim>
```

Challenges:

- ☐ Couldn't differentiate between `eval()` and `exec()` clearly at first — led to wrong usage.
- ☐ Struggled to display proper error messages when code execution failed.

- ❑ Faced issue when users entered input with syntax errors — caused full program crash without traceback.

Task 2 :

Code Snippet & Screenshot

#get maths expression from user

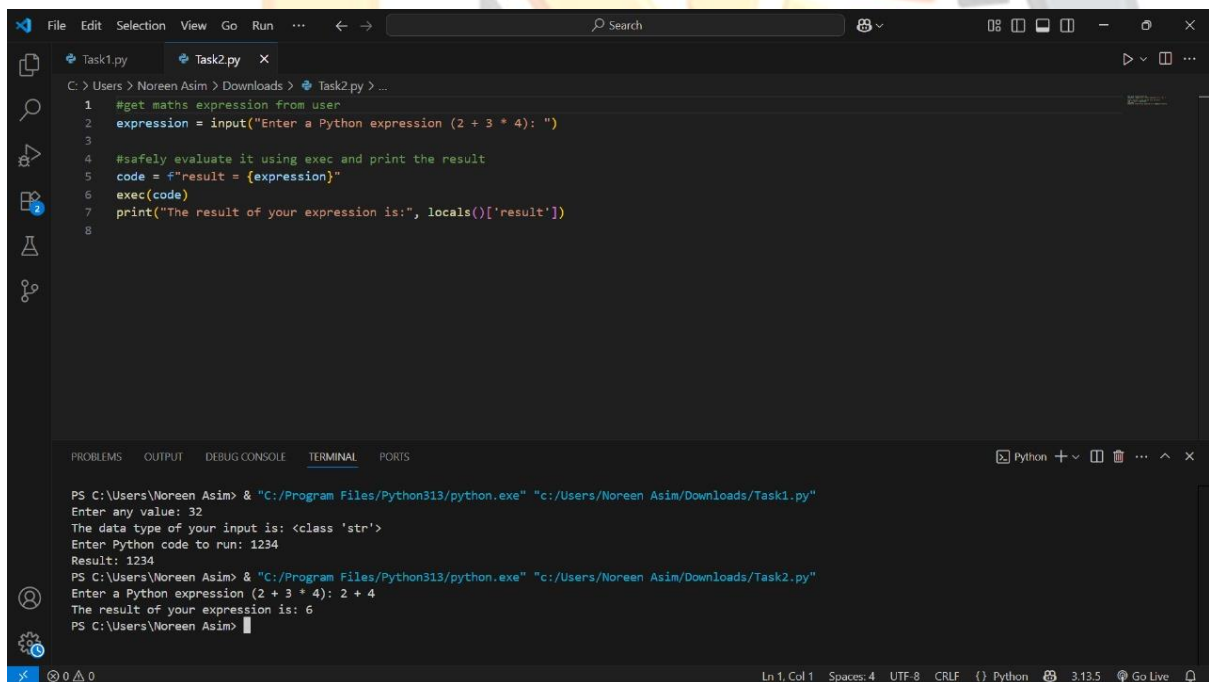
```
expression = input("Enter a Python expression (2 + 3 * 4): ")
```

#safely evaluate it using exec and print the result

```
code = f"result = {expression}"
```

```
exec(code)
```

```
print("The result of your expression is:", locals()['result'])
```



The screenshot shows a Python IDE with a dark theme. The editor window displays the code snippet from the previous blocks. Below the editor, the terminal window shows the execution of the program. The prompt 'PS C:\Users\Noreen Asim>' is followed by the command to run 'Task1.py' and then 'Task2.py'. The user enters '32' and '1234' for the first run, and '2 + 4' for the second run. The output shows the data type of the input and the result of the evaluation.

```
PS C:\Users\Noreen Asim> "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task1.py"
Enter any value: 32
The data type of your input is: <class 'str'>
Enter Python code to run: 1234
Result: 1234

PS C:\Users\Noreen Asim> "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task2.py"
Enter a Python expression (2 + 3 * 4): 2 + 4
The result of your expression is: 6
PS C:\Users\Noreen Asim>
```

Challenges:

- ❑ Didn't initially validate the user's expression for malicious code (e.g., `os.system()`).
- ❑ Difficulty understanding how `locals()` retrieves dynamically created variables.
- ❑ Forgot to wrap `exec()` inside try-except for safer error handling.

Task 3 :

Code Snippet & Screenshot

```
import math
```

```
radius = float(input("Enter radius of the circle: "))
```

```
#pow = power, pi = 3.14.....
```

```
area = math.pi * math.pow(radius, 2)
```

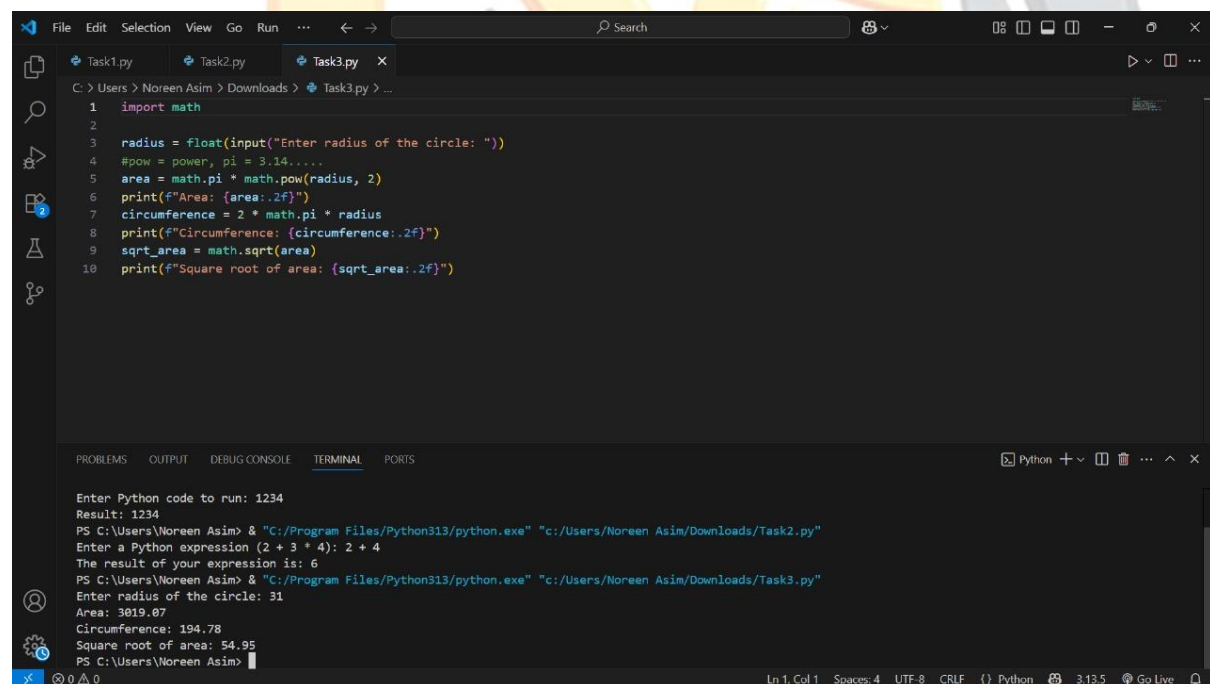
```
print(f"Area: {area:.2f}")
```

```
circumference = 2 * math.pi * radius
```

```
print(f"Circumference: {circumference:.2f}")
```

```
sqrt_area = math.sqrt(area)
```

```
print(f"Square root of area: {sqrt_area:.2f}")
```



The screenshot shows a Python IDE with a dark theme. The editor window displays the code snippet from the previous blocks. The terminal window at the bottom shows the execution of the code. It starts with a prompt to enter Python code to run, followed by the execution of Task2.py and Task3.py. The output for Task3.py shows the results of the calculations for a radius of 31.

```
Enter Python code to run: 1234
Result: 1234
PS C:\Users\Noreen Asim> & "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task2.py"
Enter a Python expression (2 + 3 * 4): 2 + 4
The result of your expression is: 6
PS C:\Users\Noreen Asim> & "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task3.py"
Enter radius of the circle: 31
Area: 3019.07
Circumference: 194.78
Square root of area: 54.95
PS C:\Users\Noreen Asim>
```

Challenges:

- ☐ Initially used `**` instead of `math.pow()` and got inconsistent results.
- ☐ Confused between `.format()` and f-strings while formatting output.
- ☐ Didn't round float input which sometimes gave long decimal outputs.

Task 4 :

Code Snippet & Screenshot

```
import random
```

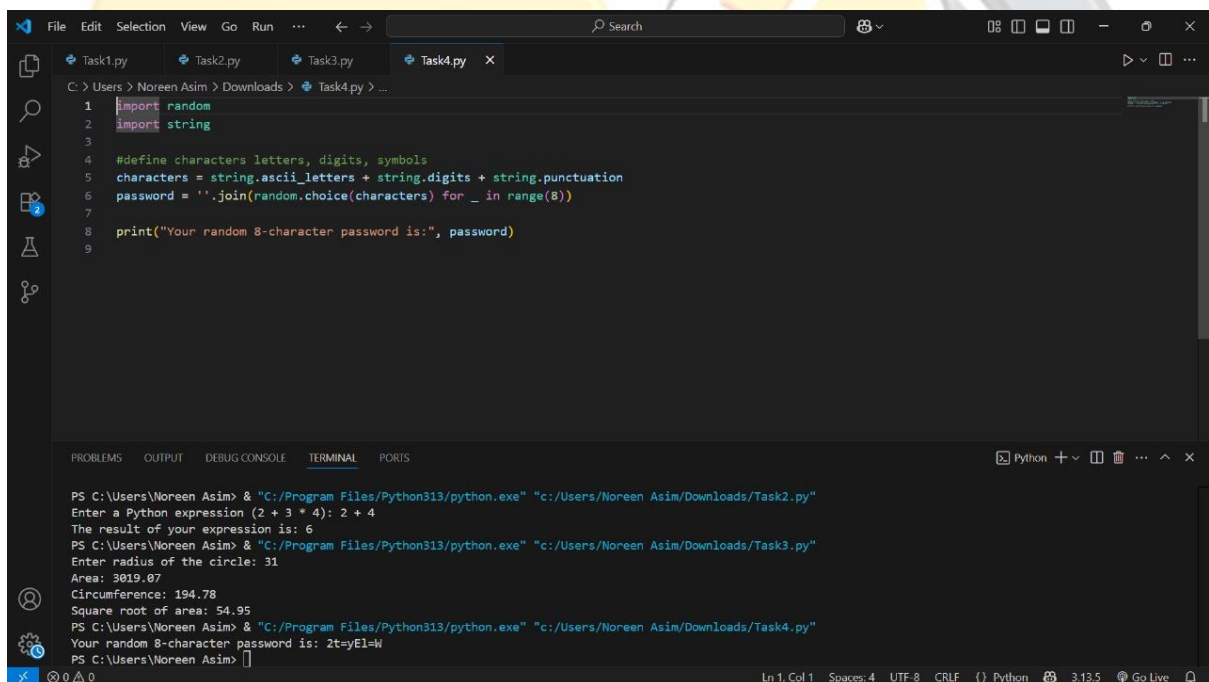
```
import string
```

```
#define characters letters, digits, symbols
```

```
characters = string.ascii_letters + string.digits + string.punctuation
```

```
password = ''.join(random.choice(characters) for _ in range(8))
```

```
print("Your random 8-character password is:", password)
```



The screenshot shows a Python IDE with a dark theme. The editor displays the code for Task 4, which generates an 8-character random password. The terminal window at the bottom shows the execution of three Python scripts: Task2.py, Task3.py, and Task4.py. The output for Task4.py shows the generated password: "2t-yE1=W".

```
1 import random
2 import string
3
4 #define characters letters, digits, symbols
5 characters = string.ascii_letters + string.digits + string.punctuation
6 password = ''.join(random.choice(characters) for _ in range(8))
7
8 print("Your random 8-character password is:", password)
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Noreen Asim> "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task2.py"
Enter a Python expression (2 + 3 * 4): 2 + 4
The result of your expression is: 6

PS C:\Users\Noreen Asim> "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task3.py"
Enter radius of the circle: 31
Area: 3019.07
Circumference: 194.78
Square root of area: 54.95

PS C:\Users\Noreen Asim> "C:/Program Files/Python313/python.exe" "c:/Users/Noreen Asim/Downloads/Task4.py"
Your random 8-character password is: 2t-yE1=W
PS C:\Users\Noreen Asim>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.13.5 Go Live

Challenges:

- ☐ Difficulty limiting symbols to avoid special characters that break usage (e.g., quotes).
- ☐ Password sometimes included unreadable characters — needed filtering.
- ☐ Trouble understanding how string concatenation and character pools work in random.choice

Task 5 :

Code Snippet & Screenshot

```
from datetime import datetime
```

```
#get user's birth date as input
```

```
birth_input = input("Enter your birth date (YYYY-MM-DD): ")
```

```
birth_date = datetime.strptime(birth_input, "%Y-%m-%d")
```

```
#get today's date
```

```
today = datetime.today()
```

```
#calculate age in years
```

```
age_years = today.year - birth_date.year
```

```
if (today.month, today.day) < (birth_date.month, birth_date.day):
```

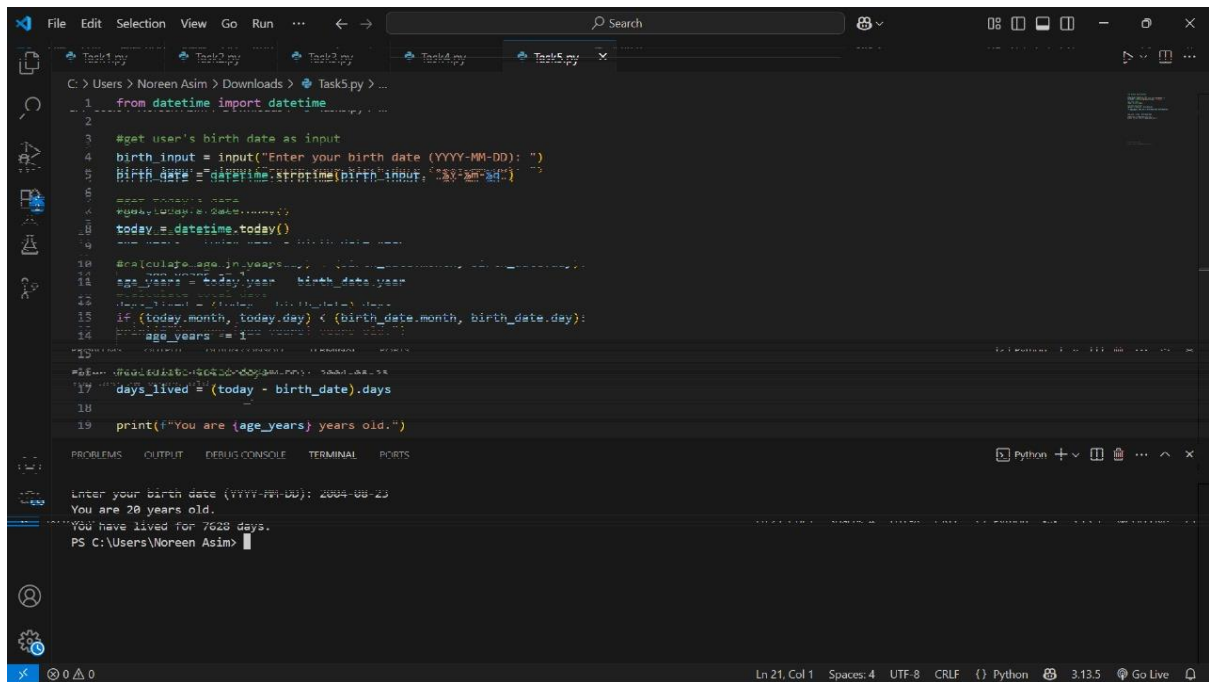
```
    age_years -= 1
```

```
#calculate total days
```

```
days_lived = (today - birth_date).days
```

```
print(f"You are {age_years} years old.")
```

```
print(f"You have lived for {days_lived} days.")
```



```
1 from datetime import datetime
2
3 #get user's birth date as input
4 birth_input = input("Enter your birth date (YYYY-MM-DD): ")
5 birth_date = datetime.strptime(birth_input, "%Y-%m-%d")
6
7 #get today's date
8 today = datetime.today()
9
10 #calculate age in years
11 age_years = today.year - birth_date.year
12
13 #check if birthday has passed this year
14 if (today.month, today.day) < (birth_date.month, birth_date.day):
15     age_years -= 1
16
17 #calculate days lived
18 days_lived = (today - birth_date).days
19
20 print(f"You are {age_years} years old.")
21
22 #get today's date
23 today = datetime.today()
24
25 #calculate days lived
26 days_lived = (today - birth_date).days
27
28 print(f>You have lived for {days_lived} days.")
```

Enter your birth date (YYYY-MM-DD): 2004-08-23
You are 20 years old.
You have lived for 7628 days.
PS C:\Users\Noreen Asim>

Challenges:

- ☐ Error when input format wasn't strictly YYYY-MM-DD; needed date validation.
- ☐ Confused by how `.strptime` works — initially used wrong format codes.
- ☐ Misunderstood date subtraction result (assumed it returned int, not timedelta).

Task 6 :

Code Snippet & Screenshot

```
import os
import re

folder = r"C:\Users\fNoreen Asim\Desktop\Important\PowerPair Sem 4\PowerPair Sem 3\Maaz"

if os.path.exists(folder):

    print("\nMatching .txt files starting with 'report':\n")

    for file in os.listdir(folder):

        if file.endswith(".txt") and file.lower().startswith("report"):

            print(file)
```

else:

```
print("Folder not found. Please check the path and try again.")
```

Challenges:

- ☐ Used `startswith("Report")` instead of `lower().startswith()` — missed files due to case sensitivity.
- ☐ Encountered `PermissionError` or `FileNotFoundError` due to incorrect folder access permissions.
- ☐ Didn't know how to filter files using regular expressions (`re.match`) properly.

