TECHNIK NEST

INNOVATIVE MINDS, NESTING SUCCESS

Name: Hassan Ali

Intern ID: TN/IN02/PY/012

Email ID: hassanali2127294@gmail.com

Task Week: 6

Internship Domain: Python Development

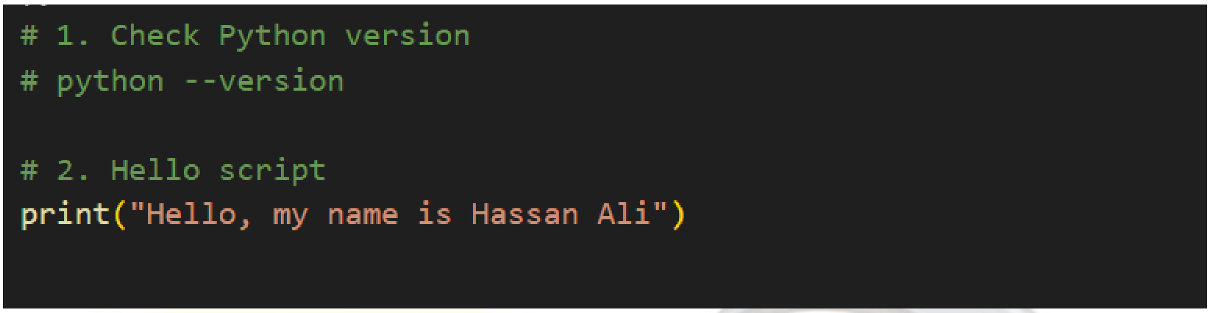
Instructor Name: Mr. Hassan Ali

Task no: 1

Install Python & print version.

Run hello script printing your name.

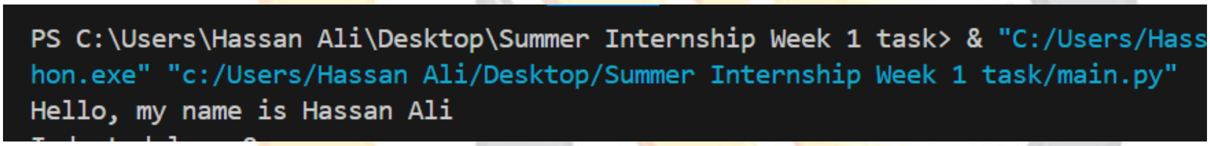
Code:



```
# 1. Check Python version
# python --version

# 2. Hello script
print("Hello, my name is Hassan Ali")
```

Output:



```
PS C:\Users\Hassan Ali\Desktop\Summer Internship Week 1 task> & "C:/Users/Hass
hon.exe" "c:/Users/Hassan Ali/Desktop/Summer Internship Week 1 task/main.py"
Hello, my name is Hassan Ali
```

Problem Faced:

Understanding how to run Python commands from the terminal may be confusing at first.

What You Learn:

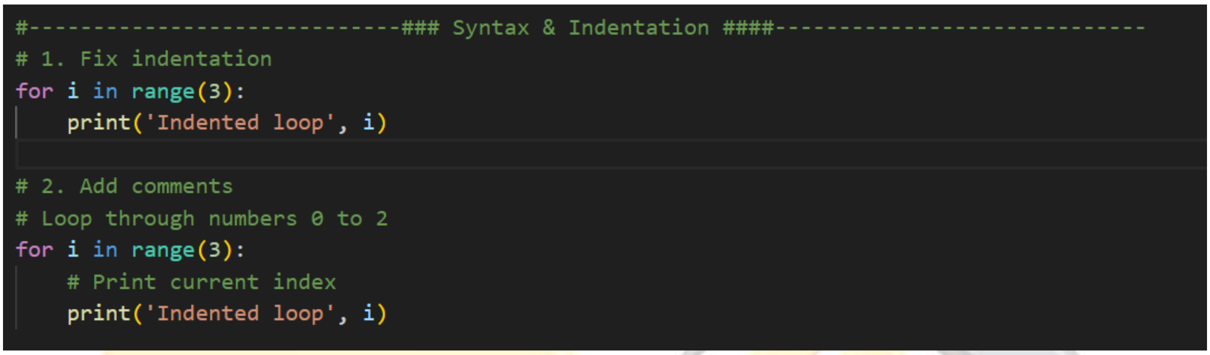
Learned how to check the Python version and create a basic print statement.

Task no: 2

Fix badly-indented code.

Add comments explaining each step.

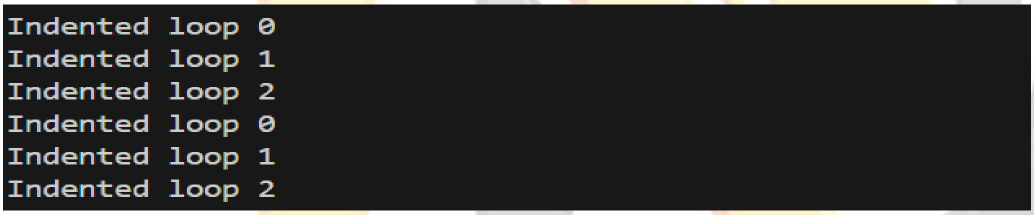
Code:



```
#-----------------------------### Syntax & Indentation ####-----------------------------
# 1. Fix indentation
for i in range(3):
    print('Indented loop', i)

# 2. Add comments
# Loop through numbers 0 to 2
for i in range(3):
    # Print current index
    print('Indented loop', i)
```

Output:



```
Indented loop 0
Indented loop 1
Indented loop 2
Indented loop 0
Indented loop 1
Indented loop 2
```

Problem Faced:

Improper indentation can cause syntax errors in Python.

What You Learn:

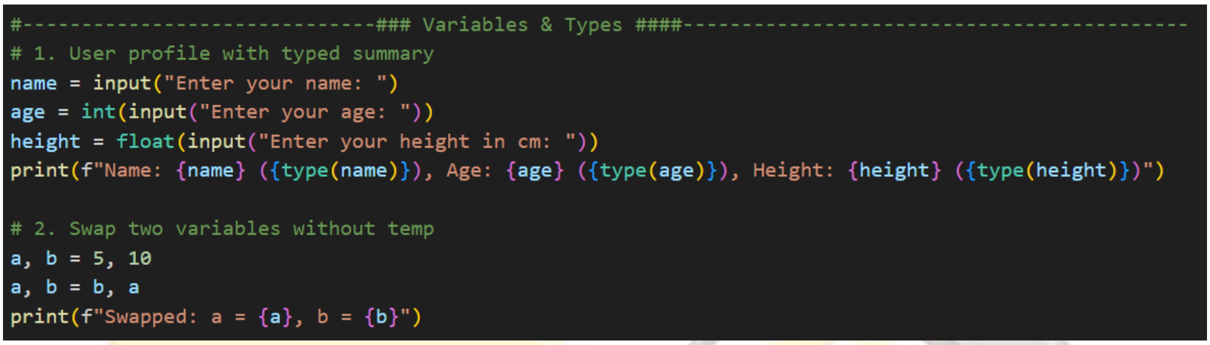
Understood the importance of indentation and how Python uses it to define code blocks like loops.

Task no: 3

Collect user profile & print typed summary.

Swap two variables without temp var.

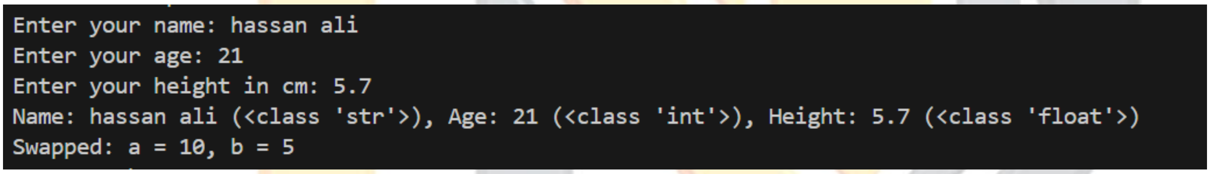
Code:



```python
#------------------------------### Variables & Types ####------------------------------------------
# 1. User profile with typed summary
name = input("Enter your name: ")
age = int(input("Enter your age: "))
height = float(input("Enter your height in cm: "))
print(f"Name: {name} ({type(name)}), Age: {age} ({type(age)}), Height: {height} ({type(height)})")

# 2. Swap two variables without temp
a, b = 5, 10
a, b = b, a
print(f"Swapped: a = {a}, b = {b}")
```

Output:



```
Enter your name: hassan ali
Enter your age: 21
Enter your height in cm: 5.7
Name: hassan ali (<class 'str'>), Age: 21 (<class 'int'>), Height: 5.7 (<class 'float'>)
Swapped: a = 10, b = 5
```

Problem Faced:

Using input() always returns a string, so converting types (like int/float) is necessary.

What You Learn:

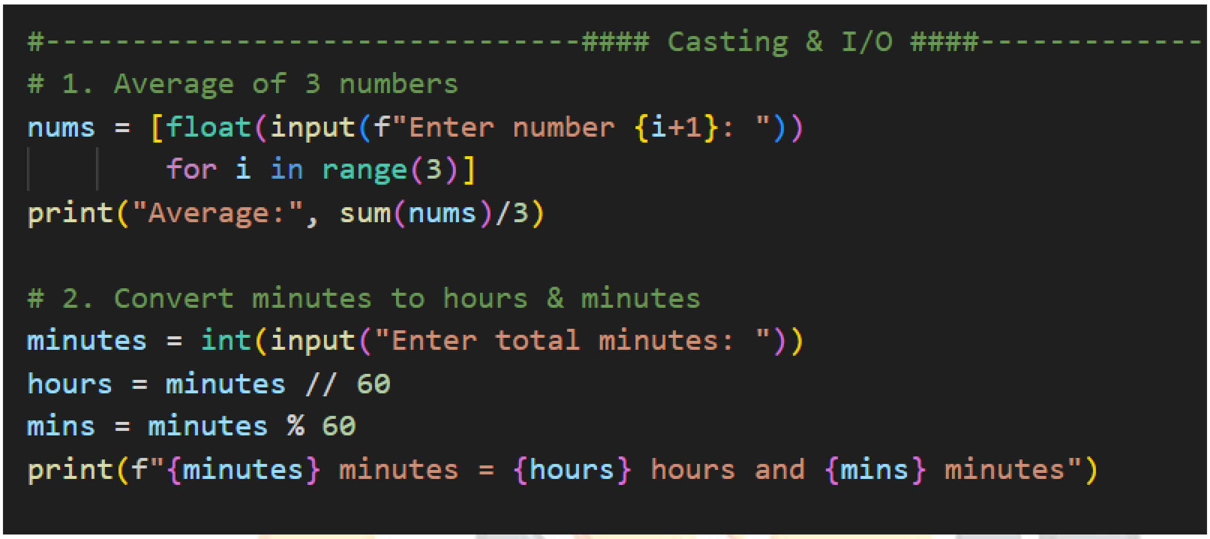
Learned how to accept user input, convert types, and use type() to check variable types.

Task no: 4

Read three numbers; output avg.

Convert minutes to hours + minutes.

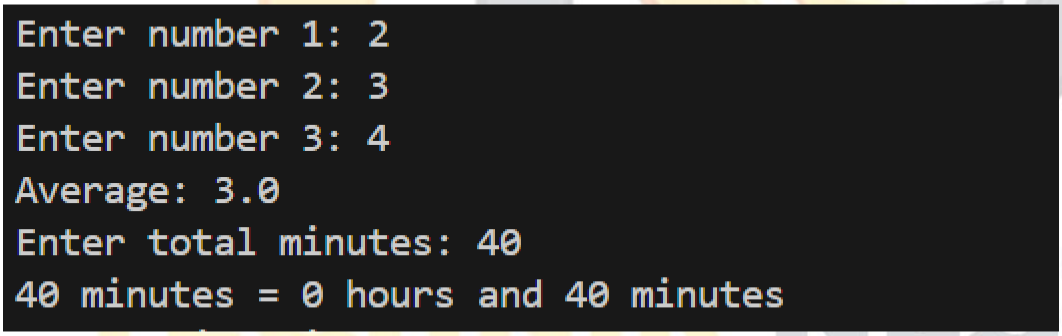
Code:



```
#--------------------------------#### Casting & I/O ####------------
# 1. Average of 3 numbers
nums = [float(input(f"Enter number {i+1}: "))
        for i in range(3)]
print("Average:", sum(nums)/3)

# 2. Convert minutes to hours & minutes
minutes = int(input("Enter total minutes: "))
hours = minutes // 60
mins = minutes % 60
print(f"{minutes} minutes = {hours} hours and {mins} minutes")
```

Output:



```
Enter number 1: 2
Enter number 2: 3
Enter number 3: 4
Average: 3.0
Enter total minutes: 40
40 minutes = 0 hours and 40 minutes
```

Problem Faced:

Forgetting to cast input strings to float can lead to incorrect calculations or errors.

What You Learn:

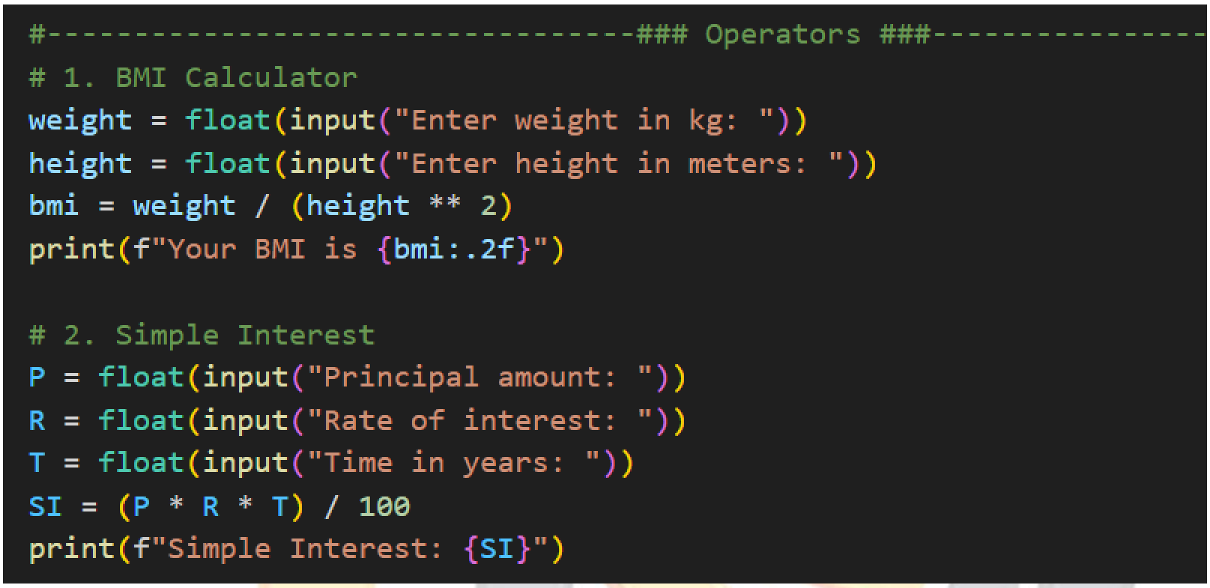
Learned how to collect multiple inputs using list comprehension and perform arithmetic operations.

Task no: 5

BMI calc from user input.

Simple interest calc.

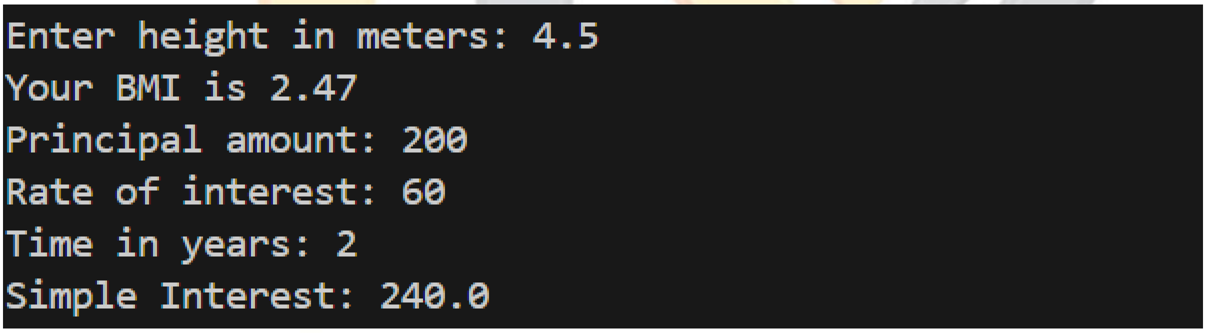
Code:



```
#----------------------------------### Operators ###----------------
# 1. BMI Calculator
weight = float(input("Enter weight in kg: "))
height = float(input("Enter height in meters: "))
bmi = weight / (height ** 2)
print(f"Your BMI is {bmi:.2f}")

# 2. Simple Interest
P = float(input("Principal amount: "))
R = float(input("Rate of interest: "))
T = float(input("Time in years: "))
SI = (P * R * T) / 100
print(f"Simple Interest: {SI}")
```

Output:



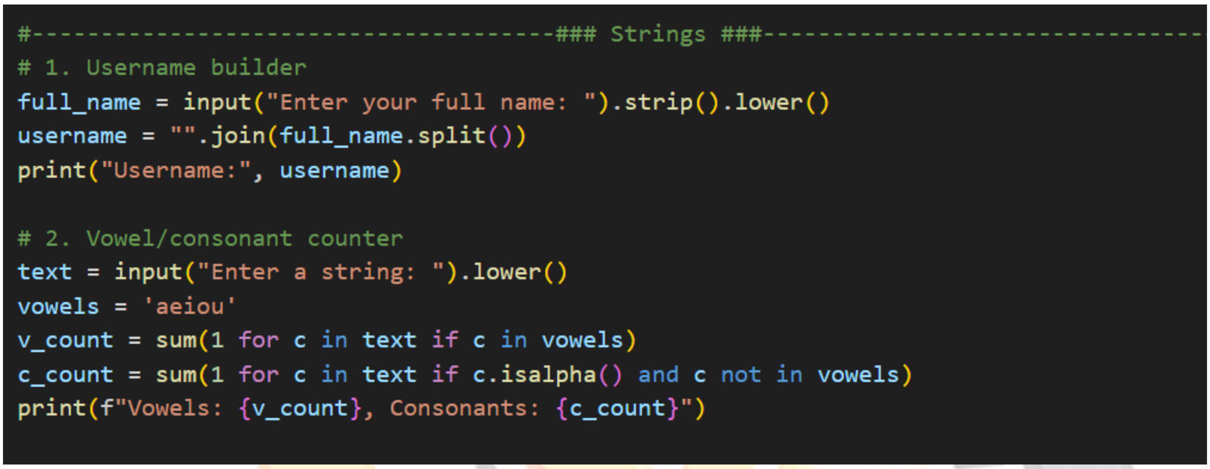
```
Enter height in meters: 4.5
Your BMI is 2.47
Principal amount: 200
Rate of interest: 60
Time in years: 2
Simple Interest: 240.0
```

Task no: 6

Username builder from full name.

Vowel/consonant counter.

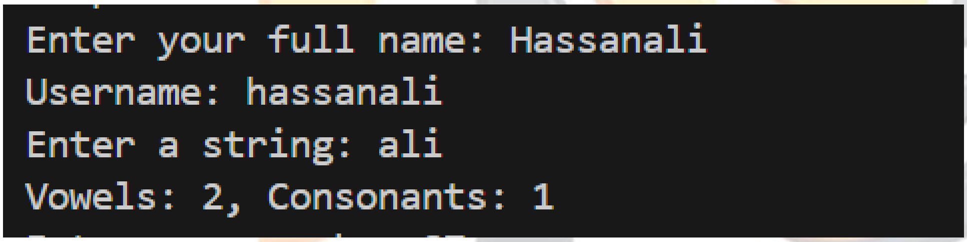
Code:



```
#------------------------------------### Strings ###-------------------------------
# 1. Username builder
full_name = input("Enter your full name: ").strip().lower()
username = "".join(full_name.split())
print("Username:", username)

# 2. Vowel/consonant counter
text = input("Enter a string: ").lower()
vowels = 'aeiou'
v_count = sum(1 for c in text if c in vowels)
c_count = sum(1 for c in text if c.isalpha() and c not in vowels)
print(f"Vowels: {v_count}, Consonants: {c_count}")
```

Output:



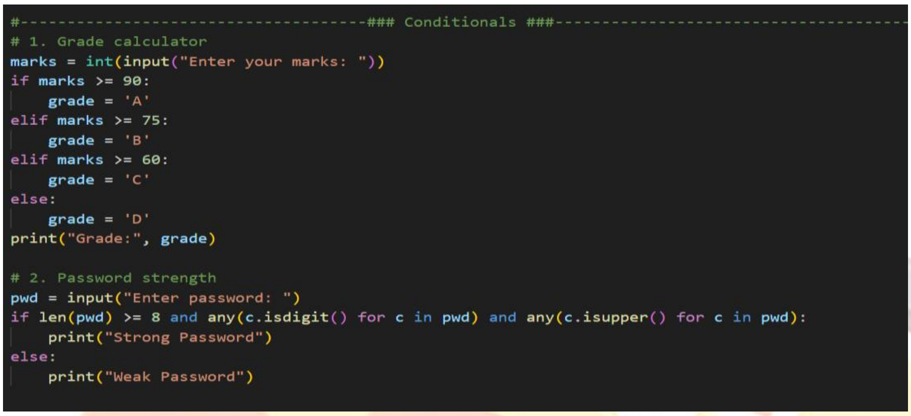
```
Enter your full name: Hassanali
Username: hassanali
Enter a string: ali
Vowels: 2, Consonants: 1
```

Task no: 7

Grade calculator.

Password strength classifier.

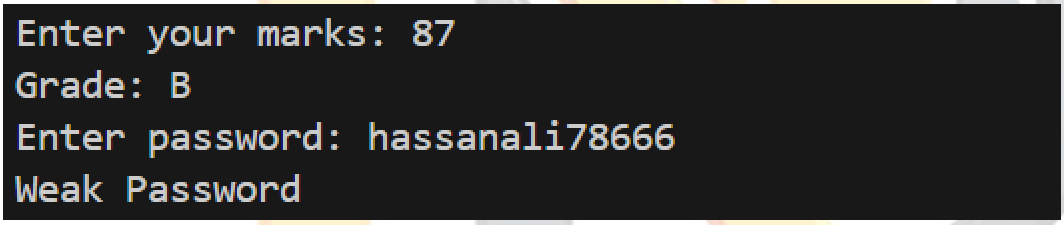
Code:



```
#------------------------------------### Conditionals ###------------------------------------
# 1. Grade calculator
marks = int(input("Enter your marks: "))
if marks >= 90:
    grade = 'A'
elif marks >= 75:
    grade = 'B'
elif marks >= 60:
    grade = 'C'
else:
    grade = 'D'
print("Grade:", grade)

# 2. Password strength
pwd = input("Enter password: ")
if len(pwd) >= 8 and any(c.isdigit() for c in pwd) and any(c.isupper() for c in pwd):
    print("Strong Password")
else:
    print("Weak Password")
```

Output:



```
Enter your marks: 87
Grade: B
Enter password: hassanali78666
Weak Password
```

Problem Faced:

It’s tricky to think of all rules for a strong password (e.g., length, digits, uppercase).

What You Learn:

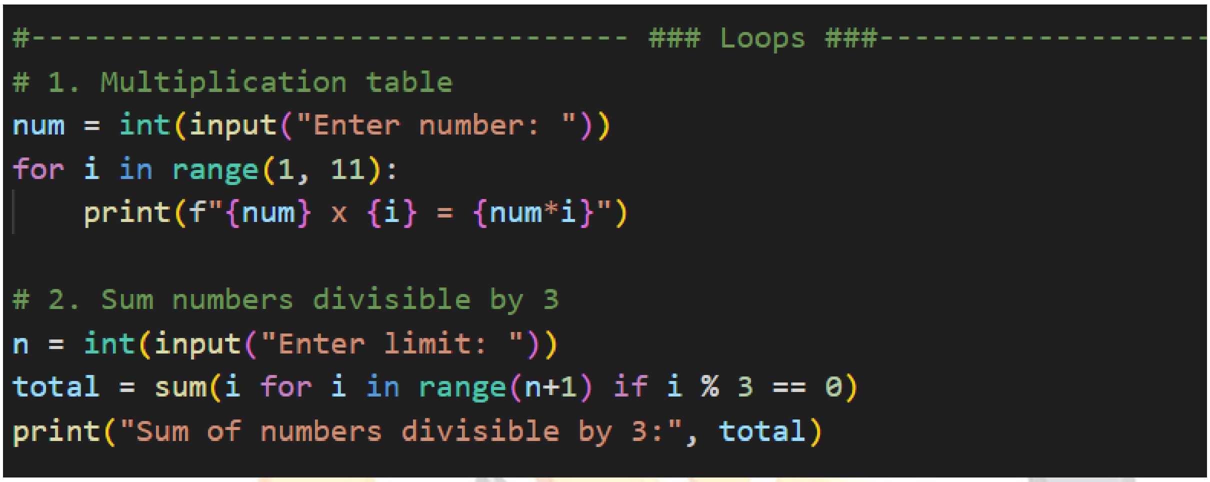
Learned how to use string methods and any() to validate password strength.

Task no: 8

Multiplication table.

Sum numbers divisible by 3.

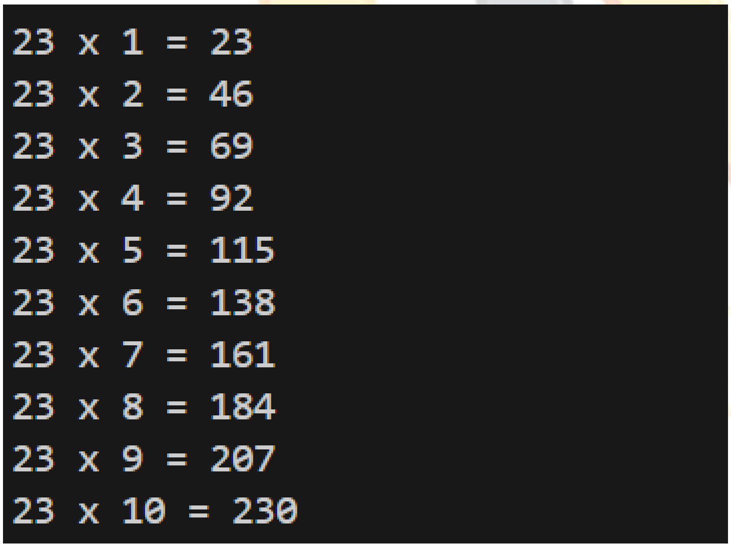
Code:



```
#-------------------------------- ### Loops ###------------------
# 1. Multiplication table
num = int(input("Enter number: "))
for i in range(1, 11):
    print(f"{num} x {i} = {num*i}")

# 2. Sum numbers divisible by 3
n = int(input("Enter limit: "))
total = sum(i for i in range(n+1) if i % 3 == 0)
print("Sum of numbers divisible by 3:", total)
```

Output:

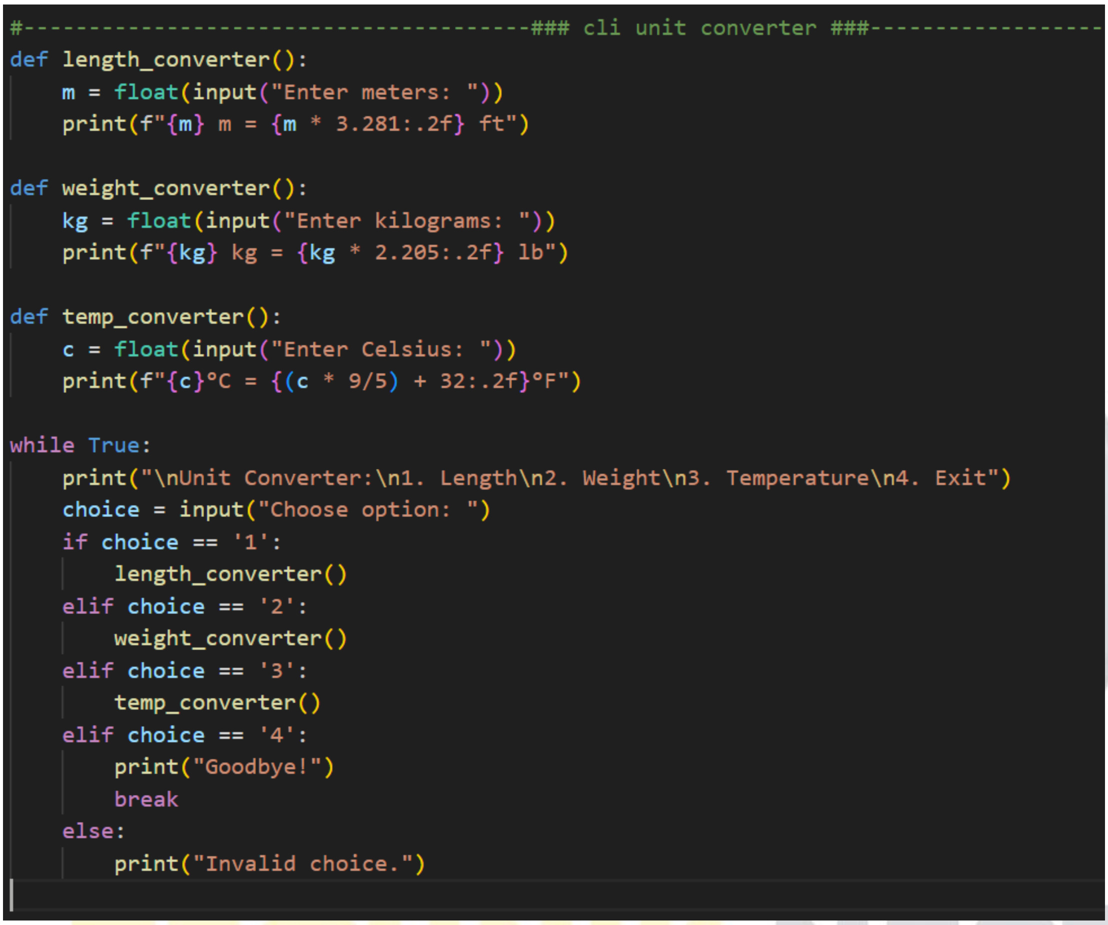


```
23 x 1 = 23
23 x 2 = 46
23 x 3 = 69
23 x 4 = 92
23 x 5 = 115
23 x 6 = 138
23 x 7 = 161
23 x 8 = 184
23 x 9 = 207
23 x 10 = 230
```

Task no: 9

CLI Unit Converter: length, weight, temperature menus + loops & conditionals.

Code:



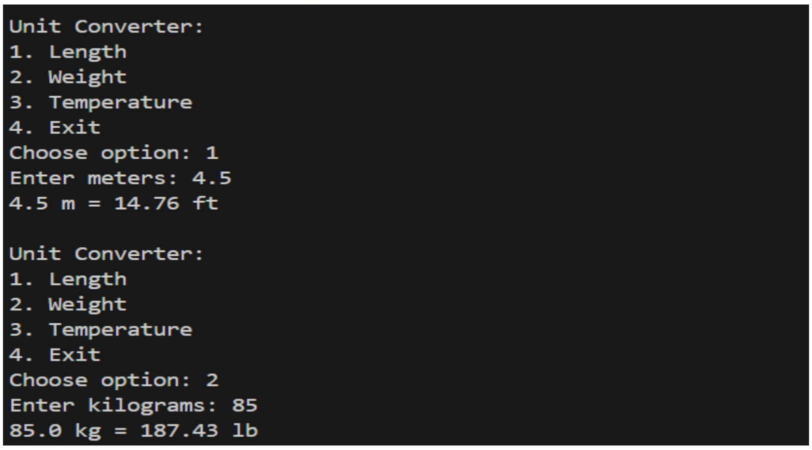
```
#---------------------------------------### cli unit converter ###-----------------
def length_converter():
    m = float(input("Enter meters: "))
    print(f"{m} m = {m * 3.281:.2f} ft")

def weight_converter():
    kg = float(input("Enter kilograms: "))
    print(f"{kg} kg = {kg * 2.205:.2f} lb")

def temp_converter():
    c = float(input("Enter Celsius: "))
    print(f"{c}°C = {(c * 9/5) + 32:.2f}°F")

while True:
    print("\nUnit Converter:\n1. Length\n2. Weight\n3. Temperature\n4. Exit")
    choice = input("Choose option: ")
    if choice == '1':
        length_converter()
    elif choice == '2':
        weight_converter()
    elif choice == '3':
        temp_converter()
    elif choice == '4':
        print("Goodbye!")
        break
    else:
        print("Invalid choice.")
```

Output:



```
Unit Converter:
1. Length
2. Weight
3. Temperature
4. Exit
Choose option: 1
Enter meters: 4.5
4.5 m = 14.76 ft

Unit Converter:
1. Length
2. Weight
3. Temperature
4. Exit
Choose option: 2
Enter kilograms: 85
85.0 kg = 187.43 lb
```

Problem Faced:

Making the menu interactive and handling wrong inputs can be confusing at first.

What You Learn:

Learned how to build a basic command-line interface using functions, loops, and conditionals.