

# **REPORT :**

**Prepared by: (SYED SAYYAM ALI SHAH)**

**Intern ID: ( TN/IN02/PY/029 )**

## **1. Hello Script**

- **Task:** Print a personalized hello message.
- **Concepts:** `print()`, functions, user input

## **2. User Profile Summary & Type Display**

- **Task:** Collect name, age, and height from user and display data type of each.
- **Concepts:** `input()`, `type()`, variables, string formatting

## **3. Swap Two Variables (No Temp Variable)**

- **Task:** Swap two integers using tuple unpacking.
- **Concepts:** Variable assignment, arithmetic or tuple unpacking

#### 4. Average of Three Numbers

- **Task:** Read 3 numbers and calculate the average.
- **Concepts:** Arithmetic, input/output

#### 5. Convert Minutes to Hours + Minutes

- **Task:** Convert a total minute value into hours and minutes.
- **Concepts:** Division, modulus operator

#### 6. BMI Calculator

- **Task:** Calculate BMI using weight and height.
- **Concepts:** Formula application, float math

#### 7. Simple Interest Calculator

- **Task:** Compute interest from principal, rate, and time.
- **Formula:** 
$$SI = P \cdot R \cdot T / 100$$
$$SI = \frac{P \cdot R \cdot T}{100}$$

## 8. Username Builder

- **Task:** Generate a username from a full name (e.g., "Sayyam Khan" → "skhan").
- **Concepts:** String manipulation, `split()`, `lower()`

## 9. Vowel and Consonant Counter

- **Task:** Count vowels and consonants in a string.
- **Concepts:** Loops, conditionals, `isalpha()`, string filtering

## 10. Grade Calculator

- **Task:** Assign grades based on numerical score.
- **Concepts:** `if-elif-else`, range checking

## 11. Password Strength Classifier

- **Task:** Classify password as Weak/Moderate/Strong.
- **Concepts:** `re` module (regex), string checks, logic conditions

## 12. Multiplication Table

- **Task:** Print multiplication table for a number.
- **Concepts:** Loops, multiplication

## 13. Sum of Numbers Divisible by 3

- **Task:** Sum numbers divisible by 3 within a range.
- **Concepts:** for loop, modulus operator, sum()

## 14. CLI Unit Converter

- **Task:** Convert between units (Length, Weight, Temperature) using a menu.
- **Concepts:** Loops, conditionals, user input, modular function design

### 💡 Skills Demonstrated:

- Input/Output Handling
- Conditional Logic
- Loops and Functions
- Data Types and Type Conversion

- Basic Arithmetic and Algorithms
- CLI Menu Navigation

## **SCREENSHOTS :**

Screenshots of all the programs are as follow:

```
def convert_length():
    print("\nLength Conversion:")
    print("1. Meters to Feet")
    print("2. Feet to Meters")
    choice = input("Choose an option (1 or 2): ")
    value = float(input("Enter value: "))

    if choice == "1":
        print(f"{value} meters = {value * 3.28084:.2f} feet")
    elif choice == "2":
        print(f"{value} feet = {value / 3.28084:.2f} meters")
    else:
        print("Invalid option.")

def convert_weight():
    print("\nWeight Conversion:")
    print("1. Kilograms to Pounds")
    print("2. Pounds to Kilograms")
    choice = input("Choose an option (1 or 2): ")
    value = float(input("Enter value: "))

    if choice == "1":
        print(f"{value} kg = {value * 2.20462:.2f} lb")
    elif choice == "2":
        print(f"{value} lb = {value / 2.20462:.2f} kg")
    else:
        print("Invalid option.")

def convert_temperature():
    print("\nTemperature Conversion:")
```

```
26     |         |         |     else:  
27     |         |         |         |     c_count += 1  
28  
29         |     print(f"Vowels: {v_count}")  
30         |     print(f"Consonants: {c_count}")  
31  
32     if __name__ == "__main__":  
33         |     username_builder()  
34         |     vowel_consonant_counter()  
35
```

--- Multiplication Table ---

Enter a number: 5

Enter the range (e.g., 10): 5

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

5 x 4 = 20

5 x 5 = 25

--- Sum of Numbers Divisible by 3 ---

Enter start of range: 1

Enter end of range: 10

Sum of numbers divisible by 3 from 1 to 10: 18

```
WEEK1\Challenge.py
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
```

```
def convert_temperature():
    print("1. Celsius to Fahrenheit")
    print("2. Fahrenheit to Celsius")
    choice = input("Choose an option (1 or 2): ")
    value = float(input("Enter temperature: "))

    if choice == "1":
        print(f"{value}°C = {((value * 9/5) + 32):.2f}°F")
    elif choice == "2":
        print(f"{value}°F = {((value - 32) * 5/9):.2f}°C")
    else:
        print("Invalid option.")

def main():
    while True:
        print("\n--- Unit Converter ---")
        print("1. Convert Length")
        print("2. Convert Weight")
        print("3. Convert Temperature")
        print("4. Exit")

        option = input("Choose an option (1-4): ")

        if option == "1":
            convert_length()
        elif option == "2":
            convert_weight()
        elif option == "3":
            convert_temperature()
        elif option == "4":
            print("Goodbye!")
            break
```

--- Grade Calculator ---

Enter your marks (0-100): 85

Your grade is: A

--- Password Strength Classifier ---

Enter your password: Passw0rd!

Password strength: Strong

```
... sting IO.py Operators.py Strings.py Conditionals.py Loops.py
Loops.py
1 def multiplication_table():
2     print("\n--- Multiplication Table ---")
3     num = int(input("Enter a number: "))
4     limit = int(input("Enter the range (e.g., 10): "))
5
6     for i in range(1, limit + 1):
7         print(f"{num} x {i} = {num * i}")
8
9 def sum_divisible_by_3():
10    print("\n--- Sum of Numbers Divisible by 3 ---")
11    start = int(input("Enter start of range: "))
12    end = int(input("Enter end of range: "))
13
14    total = sum(i for i in range(start, end + 1) if i % 3 == 0)
15    print(f"Sum of numbers divisible by 3 from {start} to {end}: {total}")
16
17 if __name__ == "__main__":
18     multiplication_table()
19     sum_divisible_by_3()
20
```

The screenshot shows a code editor window titled "INTERN\_TASK1". The file being edited is "# hello.py". The code is as follows:

```
1 # hello.py
2
3 def main():
4     print("Hello, my name is SAYYAM.")
5
6 if __name__ == "__main__":
7     main()
8
```

← → INTERN\_TASK1

Syntax And Indentation.py

```
1     print("Enter your name: ")
2     my_name = input()
3
4     print(f"Hello, my name is {my_name}.")
5
6     print("I because:")
7     print("the main() function are not indented.")
8
9     if __name__ == "__main__":
10        print("Under if __name__ == '__main__' is also not indented.")
11
12    main() are indented with 4 spaces.
13
14    under the if __name__ == "__main__" block is also indented correctly.
15    inded code will be:
16
17
18
19    print("Enter your name: ")
20    name = input()
21
22    print(f"Hello, my name is {name}.")
23
24    __main__:
```

Ln 15, Col 86 Spaces: 4 UTF-8 CRLF {} Python ⚙ A 20

```
</> Python Run main.py
1 def average_of_three():
2     print("\n--- Average of Three Numbers ---")
3     a = float(input("10: "))
4     b = float(input("20: "))
5     c = float(input("30: "))
6
7     avg = (a + b + c) / 3
8     print(f"Average: {avg}")
9
10 def convert_minutes():
11     print("\n--- Convert Minutes to Hours and Minutes ---")
12     total_minutes = int(input("Enter total minutes: "))
13
14     hours = total_minutes // 60
15     minutes = total_minutes % 60
16
17     print(f"{total_minutes} minutes = {hours} hour(s) and {minutes} minute(s)")
18
19 if __name__ == "__main__":
20     average_of_three()
21     convert_minutes()
```

*--- Average of Three Numbers ---*

Enter first number: 10

Enter second number: 20

Enter third number: 30

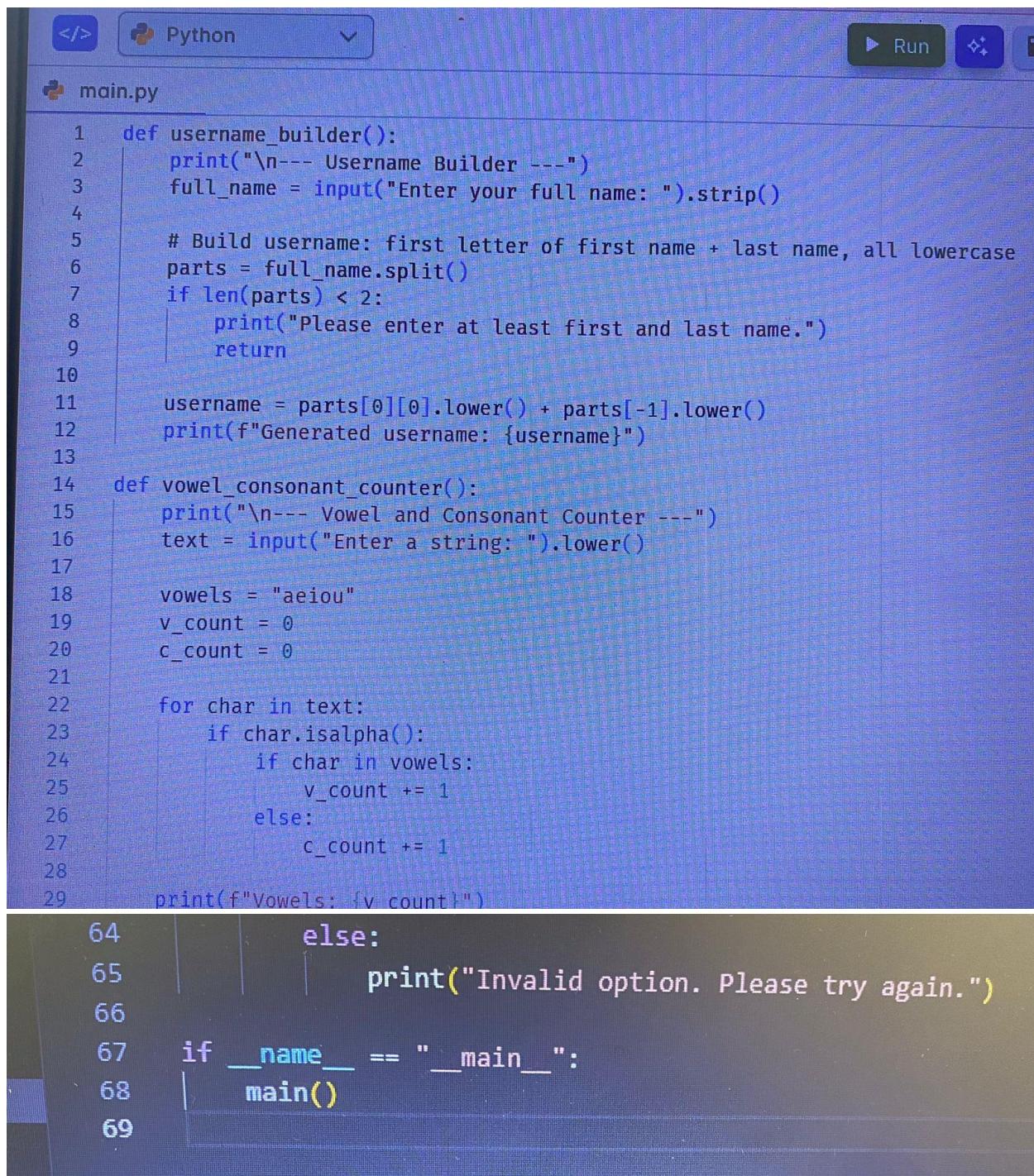
Average: 20.0



*--- Convert Minutes to Hours and Minutes ---*

Enter total minutes: 135

135 minutes = 2 hour(s) and 15 minute(s)



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

- Title Bar:** The title bar displays "</> Python" and a dropdown menu.
- Run Button:** A "Run" button with a play icon is visible in the top right corner.
- Code Area:** The main area contains the following Python code:

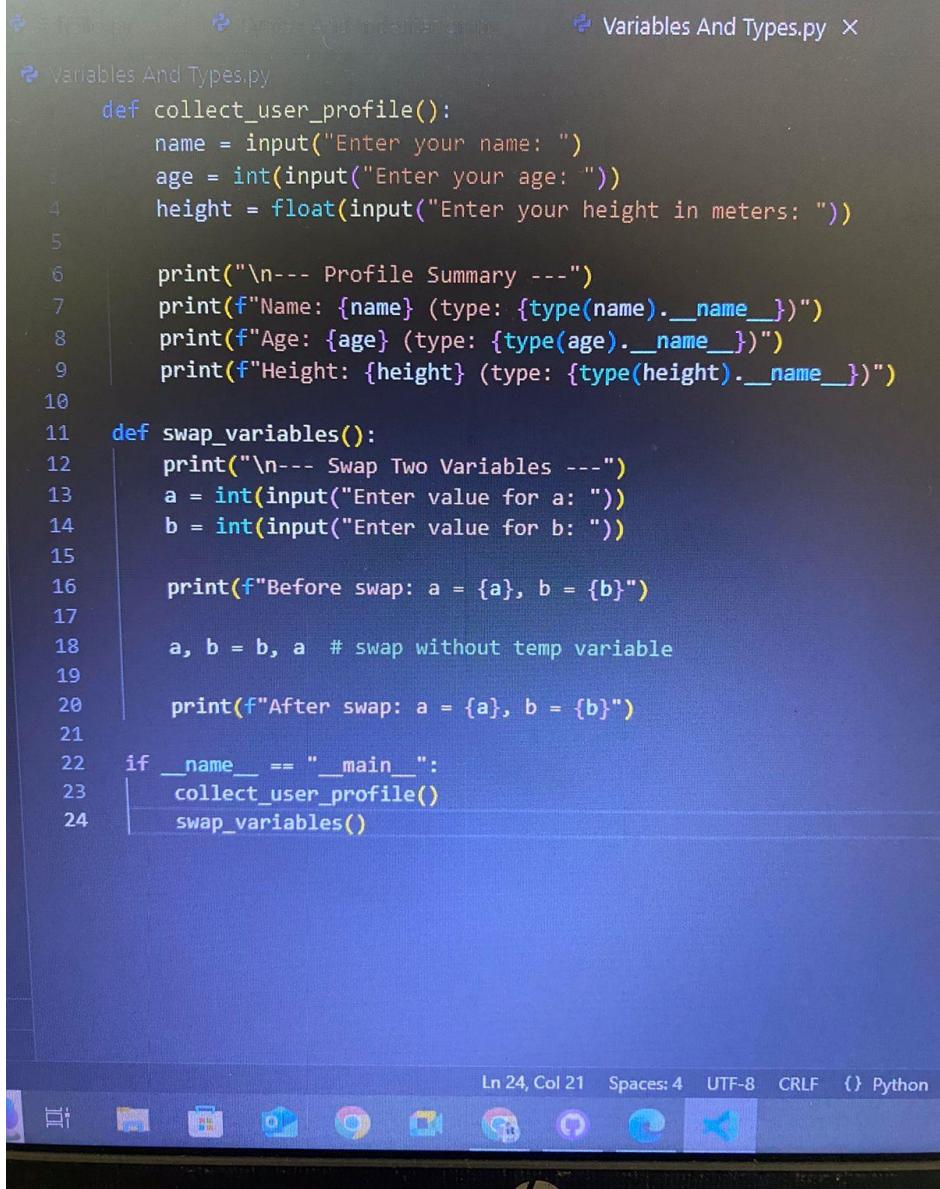
```
1  def username_builder():
2      print("\n--- Username Builder ---")
3      full_name = input("Enter your full name: ").strip()
4
5      # Build username: first letter of first name + last name, all lowercase
6      parts = full_name.split()
7      if len(parts) < 2:
8          print("Please enter at least first and last name.")
9          return
10
11     username = parts[0][0].lower() + parts[-1].lower()
12     print(f"Generated username: {username}")
13
14 def vowel_consonant_counter():
15     print("\n--- Vowel and Consonant Counter ---")
16     text = input("Enter a string: ").lower()
17
18     vowels = "aeiou"
19     v_count = 0
20     c_count = 0
21
22     for char in text:
23         if char.isalpha():
24             if char in vowels:
25                 v_count += 1
26             else:
27                 c_count += 1
28
29     print(f"Vowels: {v_count}")
30
31     else:
32         print("Invalid option. Please try again.")
33
34 if __name__ == "__main__":
35     main()
36
37
```

← → INTERN\_TASK1

Variables And Types.py X

```
Variables And Types.py
1 def collect_user_profile():
2     name = input("Enter your name: ")
3     age = int(input("Enter your age: "))
4     height = float(input("Enter your height in meters: "))
5
6     print("\n--- Profile Summary ---")
7     print(f"Name: {name} (type: {type(name).__name__})")
8     print(f"Age: {age} (type: {type(age).__name__})")
9     print(f"Height: {height} (type: {type(height).__name__})")
10
11 def swap_variables():
12     print("\n--- Swap Two Variables ---")
13     a = int(input("Enter value for a: "))
14     b = int(input("Enter value for b: "))
15
16     print(f"Before swap: a = {a}, b = {b}")
17
18     a, b = b, a # swap without temp variable
19
20     print(f"After swap: a = {a}, b = {b}")
21
22 if __name__ == "__main__":
23     collect_user_profile()
24     swap_variables()
```

Ln 24, Col 21 Spaces: 4 UTF-8 CRLF {} Python



The screenshot shows a Python code editor interface with a terminal window. The code in main.py defines two functions: bmi\_calculator() and simple\_interest\_calculator(). The bmi\_calculator function prompts for weight and height, calculates BMI, and prints the result. The simple\_interest\_calculator function prompts for principal, rate, and time, calculates simple interest, and prints the result. The terminal window shows the execution of the script, with user inputs and the resulting output for both calculators.

```
main.py
1 def bmi_calculator():
2     print("\n--- BMI Calculator ---")
3     weight = float(input("Enter your weight in kg: "))
4     height = float(input("Enter your height in meters: "))
5
6     bmi = weight / (height ** 2)
7     print(f"Your BMI is: {bmi:.2f}")
8
9 def simple_interest_calculator():
10    print("\n--- Simple Interest Calculator ---")
11    principal = float(input("Enter the principal amount: "))
12    rate = float(input("Enter the interest rate (in %): "))
13    time = float(input("Enter the time (in years): "))
14
15    interest = (principal * rate * time) / 100
16    print(f"Simple Interest: {interest:.2f}")
17
18 if __name__ == "__main__":
19     bmi_calculator()
20     simple_interest_calculator()

TERMINAL
--- BMI Calculator ---
Enter your weight in kg: 50
Enter your height in meters: 5.8
Your BMI is: 1.49

--- Simple Interest Calculator ---
Enter the principal amount: 10000
Enter the interest rate (in %): 5
Enter the time (in years): 2000
Simple Interest: 1000000.00

** Process exited - Return Code: 0 **
```

```
import re

def grade_calculator():
    print("\n--- Grade Calculator ---")
    marks = float(input("Enter your marks (0-100): "))

    if marks < 0 or marks > 100:
        print("Invalid marks entered.")
        return

    if marks >= 90:
        grade = "A+"
    elif marks >= 80:
        grade = "A"
    elif marks >= 70:
        grade = "B"
    elif marks >= 60:
        grade = "C"
    elif marks >= 50:
        grade = "D"
    else:
        grade = "F"

    print(f"Your grade is: {grade}")

def password_strength(password):
    length = len(password)
    strength = 0

    # Check length
    if length >= 8:
        strength += 1
```

```
1  def password_strength(password):
2      if length >= 8:
3          strength += 1
4
5      if re.search(r'[a-z]', password):
6          strength += 1
7      # Has uppercase
8      if re.search(r'[A-Z]', password):
9          strength += 1
10     # Has digit
11     if re.search(r'\d', password):
12         strength += 1
13     # Has special char
14     if re.search(r'[@#$%^&*(),.?":{}|<>]', password):
15         strength += 1
16
17     if strength <= 2:
18         return "Weak"
19     elif strength == 3 or strength == 4:
20         return "Moderate"
21     else:
22         return "Strong"
23
24
25 def password_strength_classifier():
26     print("\n--- Password Strength Classifier ---")
27     pwd = input("Enter your password: ")
28     result = password_strength(pwd)
29     print(f"Password strength: {result}")
30
31
32 if __name__ == "__main__":
33     grade_calculator()
34     password_strength_classifier()
```

The screenshot shows a Python development environment with a code editor and a terminal window.

**Code Editor (main.py):**

```
1 def collect_user_profile():
2     name = input("Enter your name: ")
3     age = int(input("Enter your age: "))
4     height = float(input("Enter your height in meters: "))
5
6     print("\n--- Profile Summary ---")
7     print(f"Name: {name} (type: {type(name).__name__})")
8     print(f"Age: {age} (type: {type(age).__name__})")
9     print(f"Height: {height} (type: {type(height).__name__})")
10
11 def swap_variables():
12     print("\n--- Swap Two Variables ---")
13     a = int(input("Enter value for a: "))
14     b = int(input("Enter value for b: "))
15
16     print(f"Before swap: a = {a}, b = {b}")
17
18     a, b = b, a # swap without temp variable
19
20     print(f"After swap: a = {a}, b = {b}")
21
22 if __name__ == "__main__":
23     collect_user_profile()
24     swap_variables()
```

**Terminal Output:**

```
Enter your name: SAYYAM
Enter your age: 22
Enter your height in meters: 6
--- Profile Summary ---
Name: SAYYAM (type: str)
Age: 22 (type: int)
Height: 6.0 (type: float)
--- Swap Two Variables ---
Enter value for a: 5
Enter value for b: 10
Before swap: a = 5, b = 10
After swap: a = 10, b = 5
** Process exited - Return Code: 0 *
```