**CS1101 – Programming Fundamentals Assignment 1**

**Student Name:** Merhawit Kahsay Gidey  
**Course:** CS1101 – Programming Fundamentals  
**Instructor:** Henry hu  
**Date:** June 21, 2025

**Part 1: Learning from Mistakes**

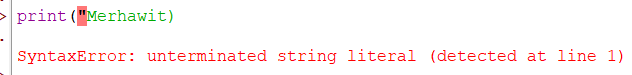
**a) Printing name with missing quotation marks**

**Code Example 1 – Missing one quotation mark:**

print("Merhawit)

**Output:**

SyntaxError: unterminated string literal (detected at line 1)

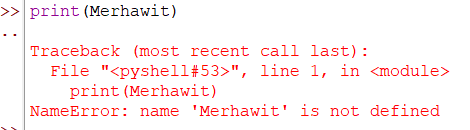


**Code Example 2 – Missing both quotation marks:**

print(Merhawit)

**Output:**

NameError: name 'Merhawit' is not defined



**Explanation:**  
In Python, string data must be enclosed in quotes (either ' or "). Missing quotation marks causes Python to either not recognize the string or think it's a variable (if both are missing), resulting in errors.

**b) Difference between \* and \*\* Operators**

**Code Example:**

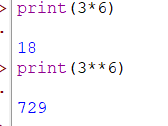
print(3 \* 6) # Multiplication

print(3 \*\* 6) # Exponentiation

**Output:**

6

9



**Explanation:**

* \* multiplies two numbers (3 times 2 = 6).
* \*\* raises a number to the power of another (3 raised to 2 = 9).

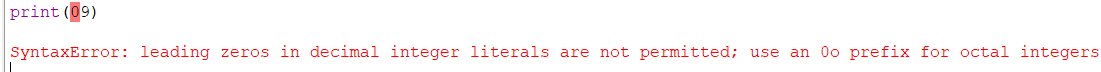
**c) Display integer like 09**

**Code Example:**

num = 09

print(num)

**Output:**

****

SyntaxError: leading zeros in decimal integer literals are not permitted; use an 0o prefix for octal integers

**Explanation:**  
Python 3 does **not allow leading zeros** in decimal numbers because they can confuse the interpreter (which might think it’s an octal number).

**d) type('67') vs type(67)**

**Code Example:**

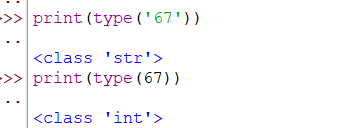
print(type('67')) # String

print(type(67)) # Integer

**Output:**

<class 'str'>

<class 'int'>



**Explanation:**

* '67' (with quotes) is a **string** type.
* 67 (without quotes) is an **integer** type.

**Part 2: Python Programs**

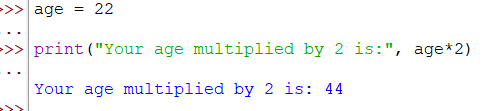
**a) Multiply age by 2**

age = 22

print("Your age multiplied by 2 is:", age \* 2)

**Output:**

Your age multiplied by 2 is: 44



**Explanation:**  
The variable age stores your age (e.g., 22). Multiplying by 2 gives 44.

**b) Display city, country, and continent**

print("City: Mekelle")

print("Country: Ethiopia")

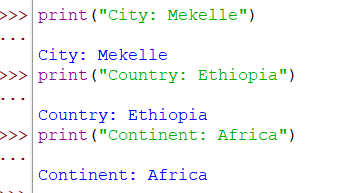
print("Continent: Africa")

**Output:**

City: Mekelle

Country: Ethiopia

Continent: Africa



**Explanation:**  
Simple string printing displays location details.

**c) Display examination schedule**

print("Examination Schedule:")

print("Start Date: July 5, 2025")

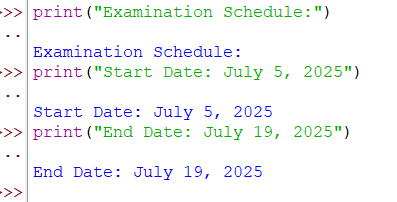
print("End Date: July 19, 2025")

**Output:**

Examination Schedule:

Start Date: July 5, 2025

End Date: July 19, 2025



**Explanation:**  
Strings provide an overview of the exam schedule.

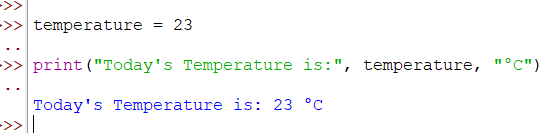
**d) Display temperature on assignment day**

temperature = 23 # Example temperature in Celsius

print("Today's Temperature is:", temperature, "°C")

**Output:**

Today's Temperature is: 23 °C



**Explanation:**  
A variable stores the day's temperature and prints it.

**Reflection**

During this assignment, I learned how **careless errors in syntax (like missing quotes)** can lead to important exceptions such as **SyntaxError and NameError**. These errors make debugging necessary and encourage writing clear and error-free code. The distinction between operators such as \* and \*\* also became clearer, where one is used for normal multiplication and the other for exponentiation, which is critical for mathematical problem solving in programming.

I also discovered that Python restricts the use of **leading zeros in integers** to prevent confusion with octal numbers, which I was not aware of before. The type() function was useful in helping me distinguish between integers and strings, reminding me that Python treats them differently in operations and memory.

In part 2, practical tasks such as multiplying an age, printing personal location details, and specifying an examination schedule gave me confidence in using **variables, strings, and basic arithmetic operators**. I also learned to display environmental data such as temperature, which reflects Python’s flexibility in solving real-life problems. Overall, this assignment helped me understand common programming mistakes, proper syntax, and the practical utility of Python for various tasks.

**References**

Downey, A. (2015). *Think Python: How to Think Like a Computer Scientist* (2nd ed.). Green Tree Press. Retrieved from <https://greenteapress.com/thinkpython2/thinkpython2.pdf>