**CS1101 – Programming Fundamentals Assignment 2**

**Student Name:** Merhawit Kahsay Gidey  
**Course:** CS1101 – Programming Fundamentals  
**Instructor:** Henry hu  
**Date:** June 28, 2025

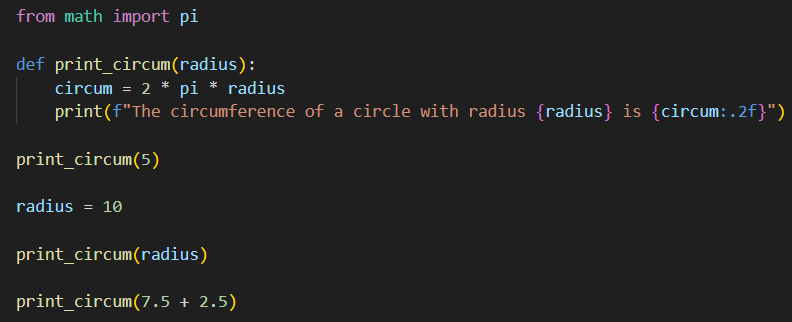
**Part 1:**

**Function Definition and Demonstration**

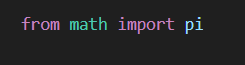
In this assignment, we are asked to create a Python function called print\_circum that calculates and prints the circumference of a circle. The formula used is:

*Circumference =2×π×radius*

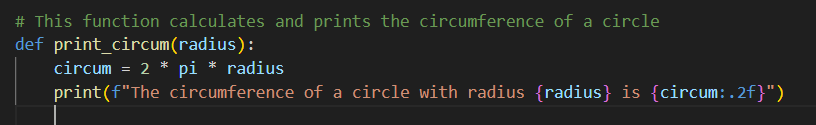
The purpose of this task is to understand how to define functions, pass arguments (both values and expressions), and perform mathematical operations using Python's built-in math module. This assignment also demonstrates how to format output using f-strings and how function calls work with different types of inputs. Below is the code for the function, three different function calls, and a detailed explanation of how each part works.

****

**Technical Explanation:**

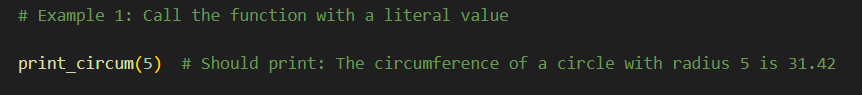
****

* This line **imports the value of π (pi)** from Python’s built-in math module.
* pi is a constant that represents the mathematical value of π, approximately 3.14159.
* Instead of writing 3.14 manually, we use pi to make our calculation **more accurate and professional**.



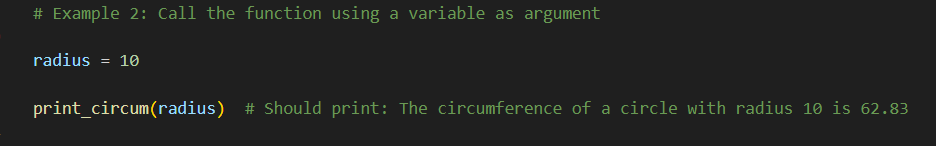
* This line **defines a function** called print\_circum.
* The word radius inside the parentheses is a **parameter**. It’s like a placeholder that will receive the actual value when the function is used.
* The function is designed to calculate the **circumference of a circle** using the value of radius.
* It uses **f-string formatting** to neatly insert values into a sentence.
* {circum:.2f} means: “Show the circum value rounded to 2 decimal places.”
* For example, if the result was 31.415926, it will print 31.42.

**Function call 1**



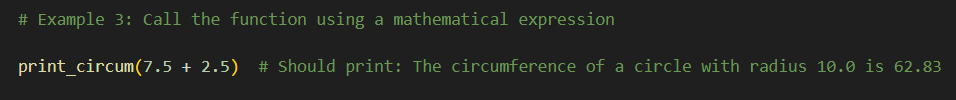
* The number 5 is passed as the **argument** for the radius.
* Inside the function, radius = 5.
* The circumference is calculated as: 2 × 3.14159 × 5 = 31.42

**Function call 2**



* Here, we first define a **variable** named radius and give it the value 10.
* Then we pass that variable to the function.
* Inside the function, radius = 10.
* Circumference is: 2 ×3.14159 × 10 = 62.832

**Function Call 3**

****

* This call uses a **math expression**: 7.5 + 2.5, which equals 10.0.
* So, it’s the same as: 2×3.14159×10.0=62.83

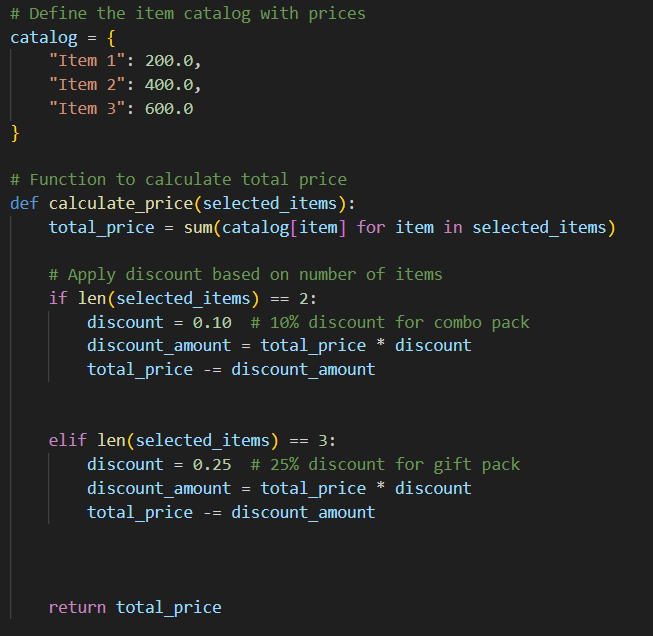
### Part 2

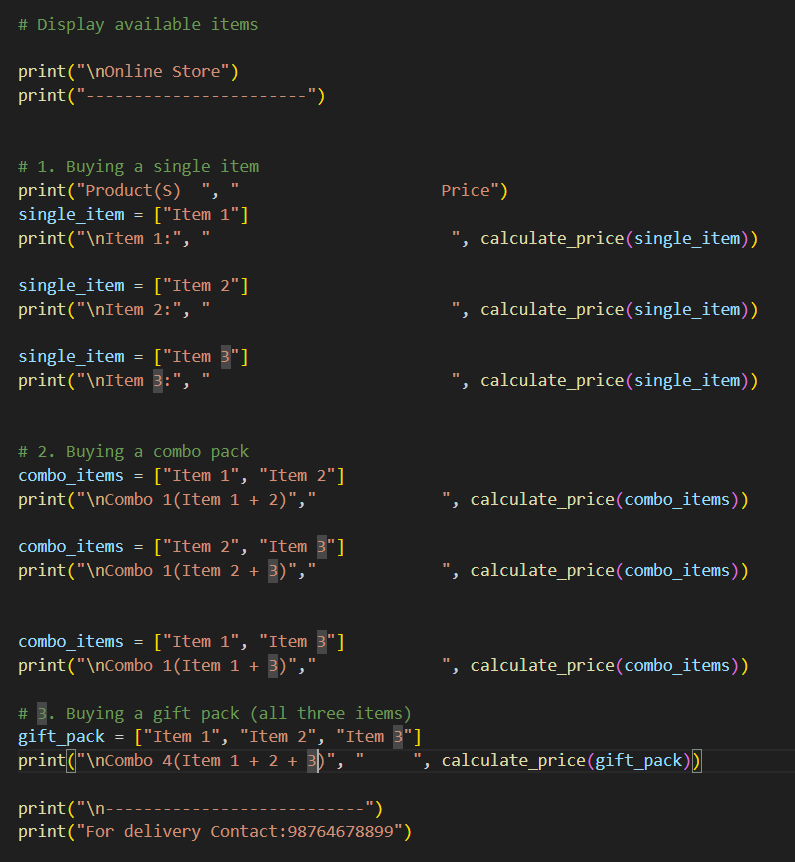
**Function Definition and Demonstration**

This function is designed to calculate the total price of one or more items selected from a catalog. It also demonstrates the use of **conditional logic**, **lists**, **loops**, and **discount application** based on the number of items selected:

* **No discount** for a single item
* **10% discount** for combo packs (2 items)
* **25% discount** for a gift pack (3 items)

So in this function I will provides a practical example of **how to build flexible and reusable code** using parameters, loops, conditionals, and formatting tools in Python.





### ****Technical Description****

* The function uses a **parameter selected\_items** which accepts a list of item names.
* The function loops through this list and **adds up the total price** from the catalog dictionary.
* A discount is applied only if:
  + **2 items** are selected → 10% discount
  + **3 items** are selected → 25% discount
* The result is then **formatted to two decimal places** using f"{value:.2f}" for better readability.
* This function shows the use of:
  + **Lists and dictionaries**
  + **Sum() function and dictionary lookup**
  + **Conditional statements**
  + **Variables for calculations**
  + **Formatted printing**

### Through this assignment, I gained hands-on experience with defining and calling functions in Python. In ****Part 1****, I learned how to perform mathematical operations using a custom function to calculate the circumference of a circle and how to pass different types of arguments such as literals, variables, and expressions. In ****Part 2****, I built a practical catalog system that applies discounts based on the number of items selected. This part helped reinforce the use of lists, dictionaries, loops, conditional logic, and formatted outputs. Overall, the assignment strengthened my understanding of how Python functions can be designed to solve real-world problems effectively and efficiently.

### ****References****

**Python Software Foundation. (n.d.-a).** Built-in functions: sum. Python.org. <https://docs.python.org/3/library/functions.html#sum>

**Python Software Foundation. (n.d.-b).** Formatted string literals (f-strings). Python.org. <https://docs.python.org/3/reference/lexical_analysis.html#f-strings>

Downey, A. (2015). Think Python: How to Think Like a Computer Scientist. [Green Tea Press](https://greenteapress.com/thinkpython2/thinkpython2.pdf). <https://greenteapress.com/thinkpython2/thinkpython2.pdf>