Benewende Pierre BONKOUNGOU

Programming Assignment Unit 2

Part 1: Circumference Calculation

In this part, the goal is to create a function called `print_circum` that takes an argument for the radius of a circle and prints the circumference of that circle. The formula for the circumference is $2\pi r$, where π is equal to 3.14159 (rounded to five decimal places). After defining this function, we will call it three times with different radius values to demonstrate its use.

Input & Output

In [30]:
```
pi = 3.14159

def print_circum(radius):
    circumference = 2 * pi * radius
    print(f"The circumference of a circle with radius {radius} is:
{circumference:.5f}")

# Call the function with different values for radius
print_circum(5)
print_circum(8.5)
print_circum(11)
```

```
The circumference of a circle with radius 5 is: 31.41590
The circumference of a circle with radius 8.5 is: 53.40703
The circumference of a circle with radius 11 is: 69.11498
```

Explanation:

- pi = 3.14159: Defines the value of π.
- `def print_circum(radius):` : Defines a function called `print_circum` that takes a radius as an argument.
- `circumference = 2 * pi * radius` : Calculates the circumference using the formula $2\pi r$.
- `print(f"The circumference of a circle with radius {radius}` is: {circumference:.5f}"): Prints the result with a formatted string showing the radius and circumference rounded to five decimal places.

- Finally, the function is called three times with different radius values to showcase its functionality.

Part 2: Company Catalog

In this part, we embark on a project where we develop a catalog for a company that sells three different items. The uniqueness lies in the fact that a customer can purchase individual items, combinations of any two items, or a gift pack containing all three items. There are special considerations for discounts, for instance, no discount for individual items, a 10% discount for combos of two items, and a 25% discount for the gift set. We need to write a function to model this scenario, perform the necessary calculations, and display the generated catalog.

Input & Output

In [32]:

```
def calculate_discount(item_count):
    """
    Parameters:
      - item_count (int): The number of items in the purchase.
    Returns:
      - float: The discount percentage.
    """
    if item_count == 1:
        return 0
    elif item_count == 2:
        return 0.10
    elif item_count == 3:
        return 0.25

def calculate_total_cost(items, item_prices):
    """
    Parameters:
      - items (list): List of items.
      - item_prices (dict): Dictionary containing the unit prices of items.
    Returns:
      - float: The total cost after applying the discount.
    """
    total_cost = sum(item_prices[item] for item in items)
    discount = calculate_discount(len(items))
    total_cost *= (1 - discount)
    return total_cost

def generate_catalog():
    # Dictionary containing the prices for individual items
    item_prices = {'Item 1': 200.0, 'Item 2': 400.0, 'Item 3': 600.0}
    # List of tuples representing different combos of items
```

```
    combos = [('Item 1', 'Item 2'), ('Item 2', 'Item 3'), ('Item 1', 'Item 3'),
('Item 1', 'Item 2', 'Item 3')]

    # Print the header of the catalog
    print("Online Store")
    print("_" * 35)
    print("{:<45} {:<10}".format("Product(s)", "Price"))

    # Print individual items and their prices
    for item, price in item_prices.items():
        print("{:<45} {:<10}".format(item, price))

    # Print combos and their total prices after applying discounts
    for i, combo in enumerate(combos, start=1):
        combo_name = f"Combo {i} ({' + '.join(combo)})"
        total_cost = calculate_total_cost(combo, item_prices)
        print("{:<45} {:<10}".format(combo_name, total_cost))

    # Print the footer of the catalog
    print("_" * 35)
    print("For delivery Contact : 98764678899")

# Example: Generate and print the catalog
generate_catalog()
```

```
Online Store
___________________________________
Product(s)                                    Price
Item 1                                        200.0
Item 2                                        400.0
Item 3                                        600.0
Combo 1 (Item 1 + Item 2)                     540.0
Combo 2 (Item 2 + Item 3)                     900.0
Combo 3 (Item 1 + Item 3)                     720.0
Combo 4 (Item 1 + Item 2 + Item 3)            900.0
___________________________________
For delivery Contact : 98764678899
```

Function Features:

- The calculate_discount function determines the discount percentage based on the number of items in a purchase. It considers three scenarios: no discount for one item, 10% for two items, and 25% for three items.
- The calculate_total_cost function computes the total cost of a purchase, factoring in the number of items, their unit price, and any applicable discount.
- The generate_catalog function assembles and prints the online store catalog. It includes individual items with their prices, various combos, and their respective total prices after applying discounts.

References:

- Downey, A. (2015). Think Python: How to think like a computer scientist. Green Tree Press. Link

- Personal ressources (Programming_Assignment_Unit-2.ipynb) https://github.com/VirtuelsDev/UoPeople/tree/598e5f42ffe18bd923b9b9bb420b2e4bd3a7991101_Programming-Fundamentals/Unit%202

In [ ]: