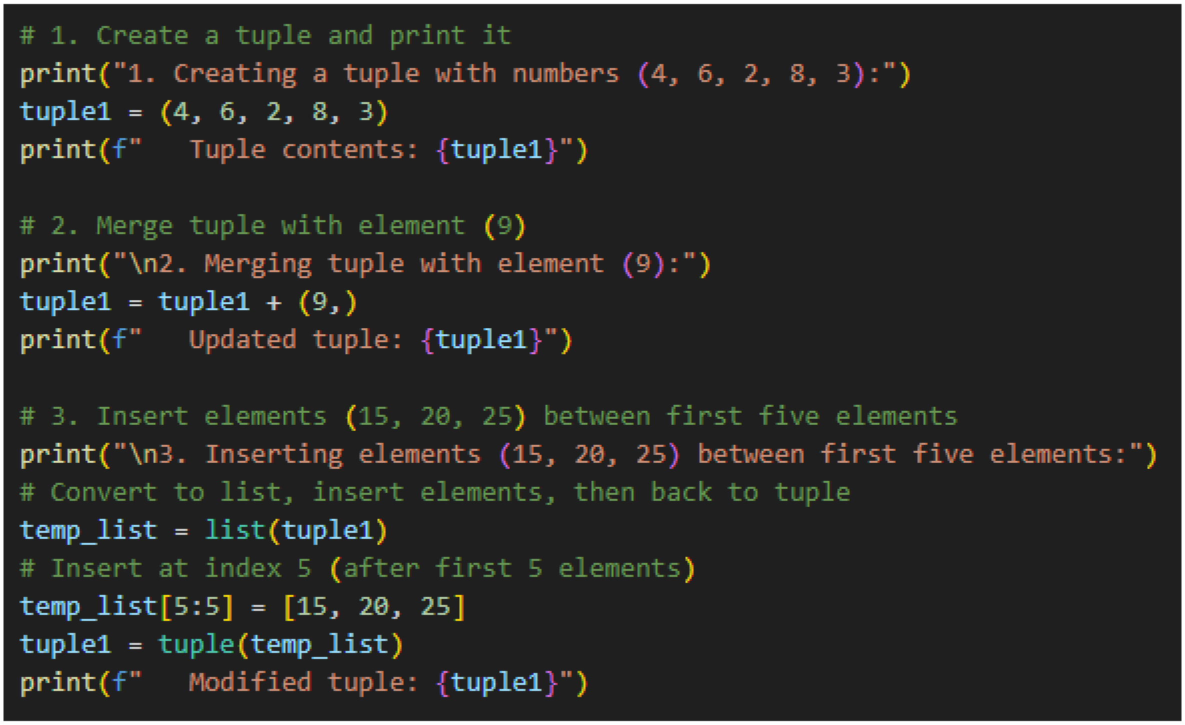
BERNARDO, Luke Joaquin - 302I

Part I-A TUPLES

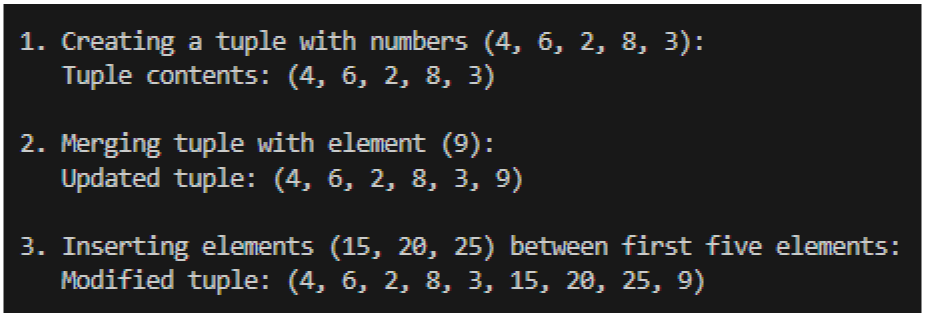
1. Create a tuple containing a sequence of numbers (4, 6, 2, 8, 3) then Print the contents of the tuple

2. Create a new tuple by merging the existing tuple with the element (9) using the + operator. Then print the updated elements of the tuple.

3. Adding items at a specific index in the tuple. Inserts elements (15, 20, 25) between the first five elements. Print the modified Tuples



```python
# 1. Create a tuple and print it
print("1. Creating a tuple with numbers (4, 6, 2, 8, 3):")
tuple1 = (4, 6, 2, 8, 3)
print(f"   Tuple contents: {tuple1}")

# 2. Merge tuple with element (9)
print("\n2. Merging tuple with element (9):")
tuple1 = tuple1 + (9,)
print(f"   Updated tuple: {tuple1}")

# 3. Insert elements (15, 20, 25) between first five elements
print("\n3. Inserting elements (15, 20, 25) between first five elements:")
# Convert to list, insert elements, then back to tuple
temp_list = list(tuple1)
# Insert at index 5 (after first 5 elements)
temp_list[5:5] = [15, 20, 25]
tuple1 = tuple(temp_list)
print(f"   Modified tuple: {tuple1}")
```



```
1. Creating a tuple with numbers (4, 6, 2, 8, 3):
   Tuple contents: (4, 6, 2, 8, 3)

2. Merging tuple with element (9):
   Updated tuple: (4, 6, 2, 8, 3, 9)

3. Inserting elements (15, 20, 25) between first five elements:
   Modified tuple: (4, 6, 2, 8, 3, 15, 20, 25, 9)
```

BERNARDO, Luke Joaquin - 302I

Part I-B TUPLES

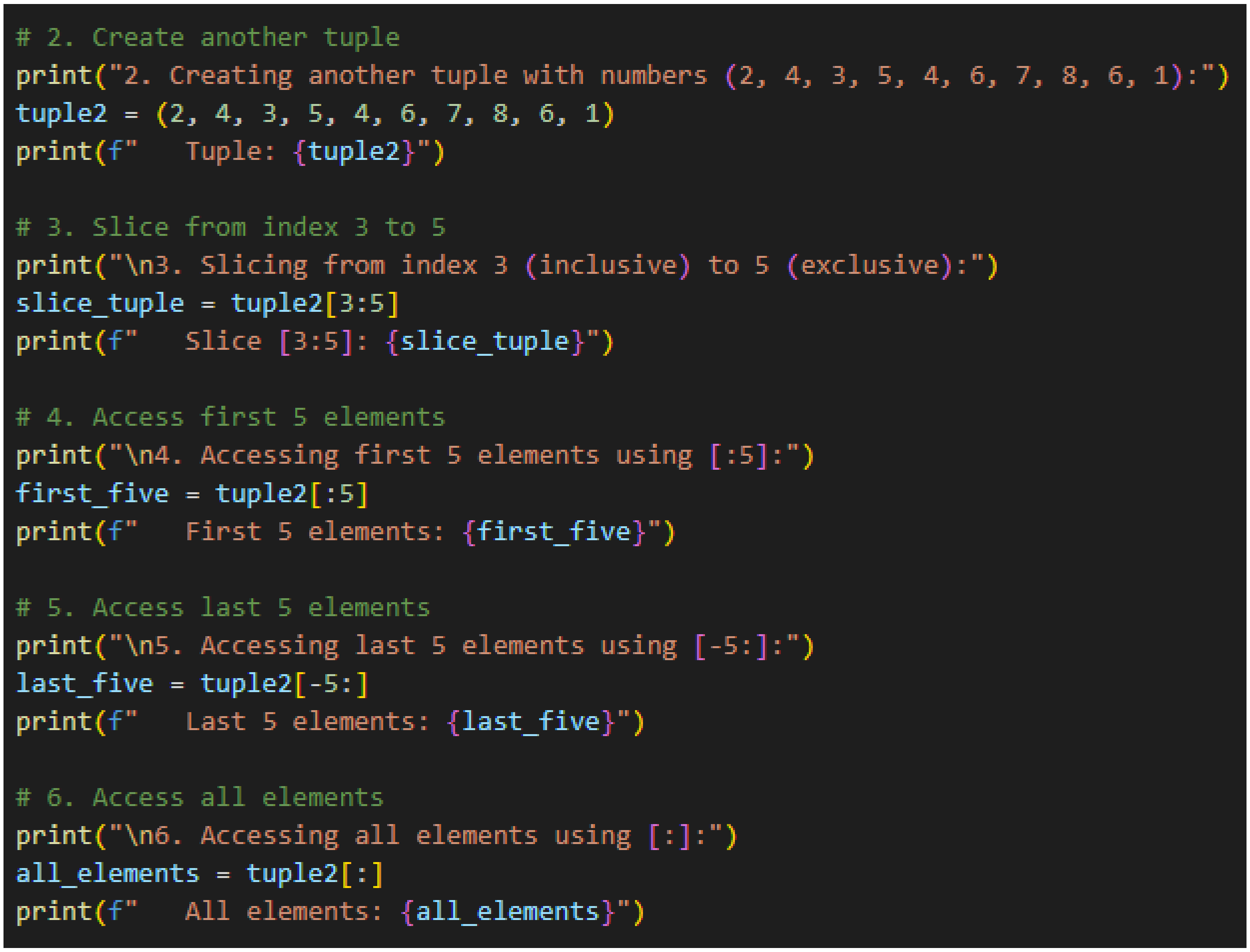
2. Create another tuple containing a sequence of numbers (2, 4, 3, 5, 4, 6, 7, 8, 6, 1)

3. Use tuple slicing (tuple[start:stop]) to extract a portion of the tuple. The start index is inclusive, and the stop index is exclusive. Slice from index 3 (inclusive) to 5 (exclusive) and store it in the variable 'slice'. Print the slice

4. If the start index isn't defined, it's taken from the beginning of the tuple. Access and Print the first 5 elements of the tuple using the [:n] format

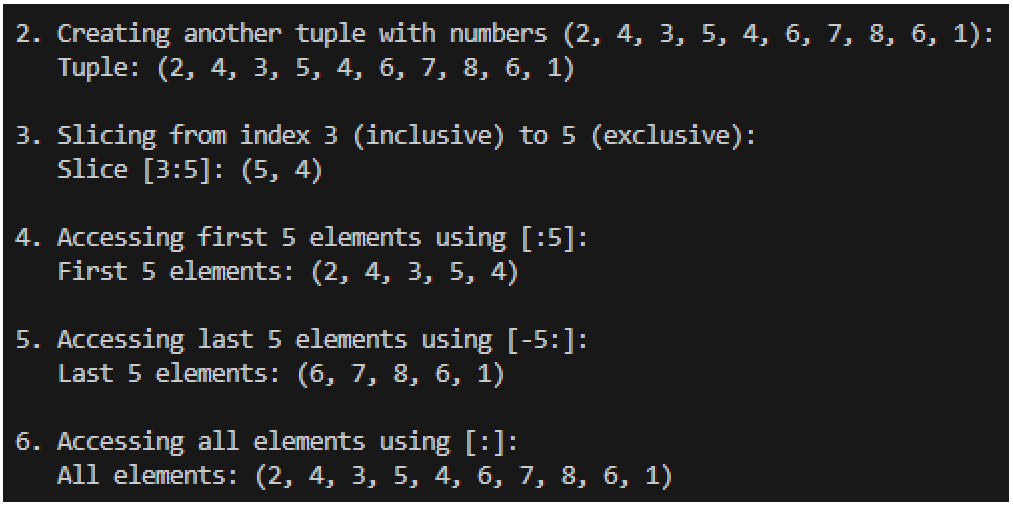
5. If the end index isn't defined, it's taken until the end of the tuple. Access and print the last 5 elements of the tuple using the [n:] format

6. If neither start nor end index is defined, it returns the full tuple. Access and print all the elements of the tuple using the [:] format



```python
# 2. Create another tuple
print("2. Creating another tuple with numbers (2, 4, 3, 5, 4, 6, 7, 8, 6, 1):")
tuple2 = (2, 4, 3, 5, 4, 6, 7, 8, 6, 1)
print(f"   Tuple: {tuple2}")

# 3. Slice from index 3 to 5
print("\n3. Slicing from index 3 (inclusive) to 5 (exclusive):")
slice_tuple = tuple2[3:5]
print(f"   Slice [3:5]: {slice_tuple}")

# 4. Access first 5 elements
print("\n4. Accessing first 5 elements using [:5]:")
first_five = tuple2[:5]
print(f"   First 5 elements: {first_five}")

# 5. Access last 5 elements
print("\n5. Accessing last 5 elements using [-5:]:")
last_five = tuple2[-5:]
print(f"   Last 5 elements: {last_five}")

# 6. Access all elements
print("\n6. Accessing all elements using [:]:")
all_elements = tuple2[:]
print(f"   All elements: {all_elements}")
```

BERNARDO, Luke Joaquin - 302I



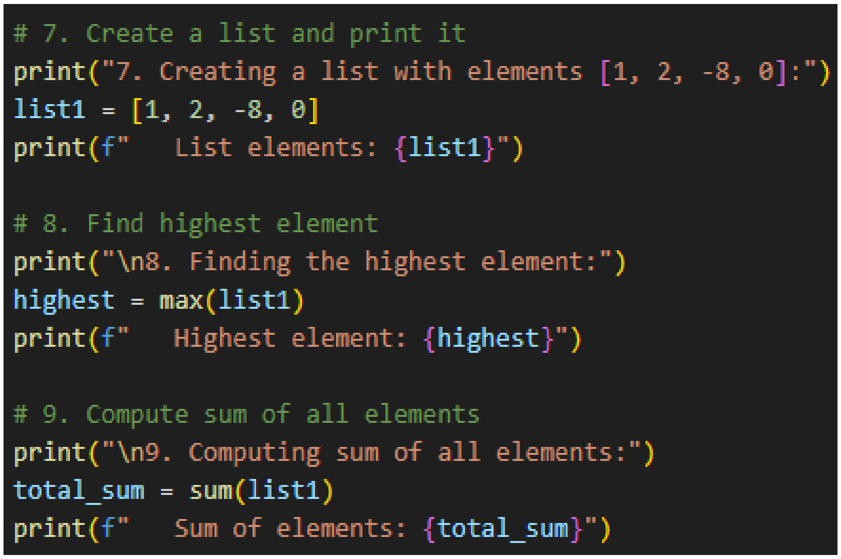
```
2. Creating another tuple with numbers (2, 4, 3, 5, 4, 6, 7, 8, 6, 1):
   Tuple: (2, 4, 3, 5, 4, 6, 7, 8, 6, 1)

3. Slicing from index 3 (inclusive) to 5 (exclusive):
   Slice [3:5]: (5, 4)

4. Accessing first 5 elements using [:5]:
   First 5 elements: (2, 4, 3, 5, 4)

5. Accessing last 5 elements using [-5:]:
   Last 5 elements: (6, 7, 8, 6, 1)

6. Accessing all elements using [:]:
   All elements: (2, 4, 3, 5, 4, 6, 7, 8, 6, 1)
```

PART II –A LIST – Reference

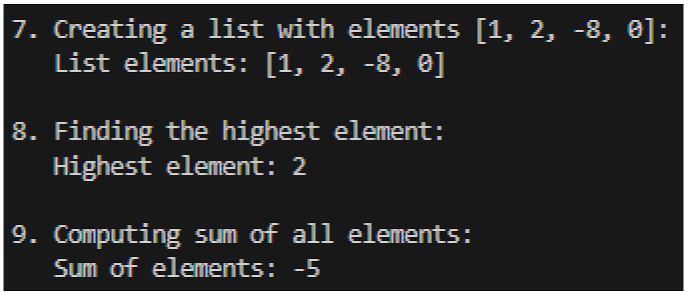
7. Create a list with elements [1, 2, -8, 0] then print all the elements of the list

8. Type a python code to determine the highest element of the created list then print the highest element.

9. Type a python code to compute the sum of all elements of the created list then print the computed sum.



```python
# 7. Create a list and print it
print("7. Creating a list with elements [1, 2, -8, 0]:")
list1 = [1, 2, -8, 0]
print(f"   List elements: {list1}")

# 8. Find highest element
print("\n8. Finding the highest element:")
highest = max(list1)
print(f"   Highest element: {highest}")

# 9. Compute sum of all elements
print("\n9. Computing sum of all elements:")
total_sum = sum(list1)
print(f"   Sum of elements: {total_sum}")
```



```
7. Creating a list with elements [1, 2, -8, 0]:
   List elements: [1, 2, -8, 0]

8. Finding the highest element:
   Highest element: 2

9. Computing sum of all elements:
   Sum of elements: -5
```

BERNARDO, Luke Joaquin - 302I

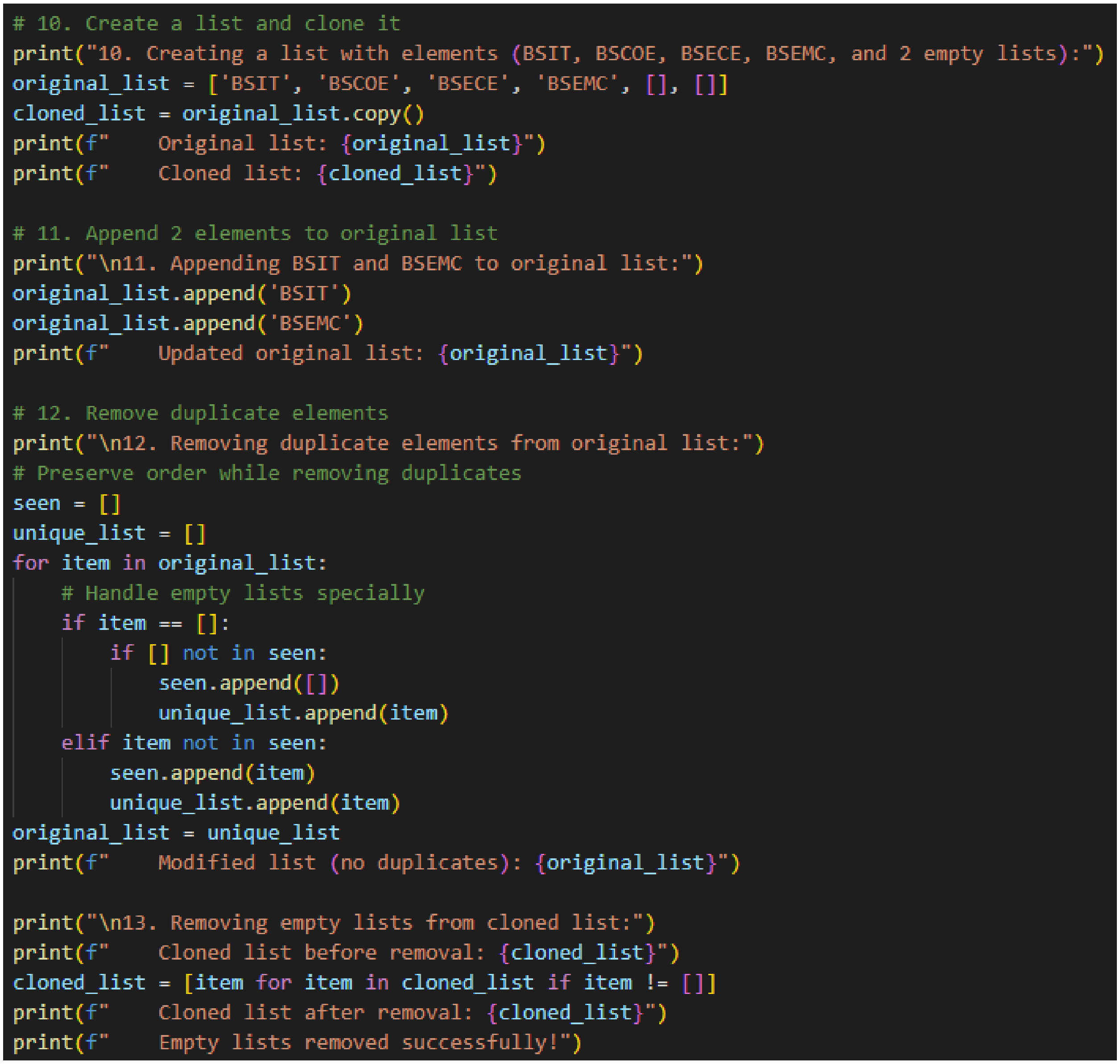
PART III – B LIST PythonListAdvanced File Lecture

10. Create a list with elements (BSIT, BSCOE, BSECE, BSEMC, and 2 empty list). Copy or clone the list created. And print both the original list and cloned list

11. On you original list use append method to add 2 elements(BSIT, BSEMC) the print the updated list

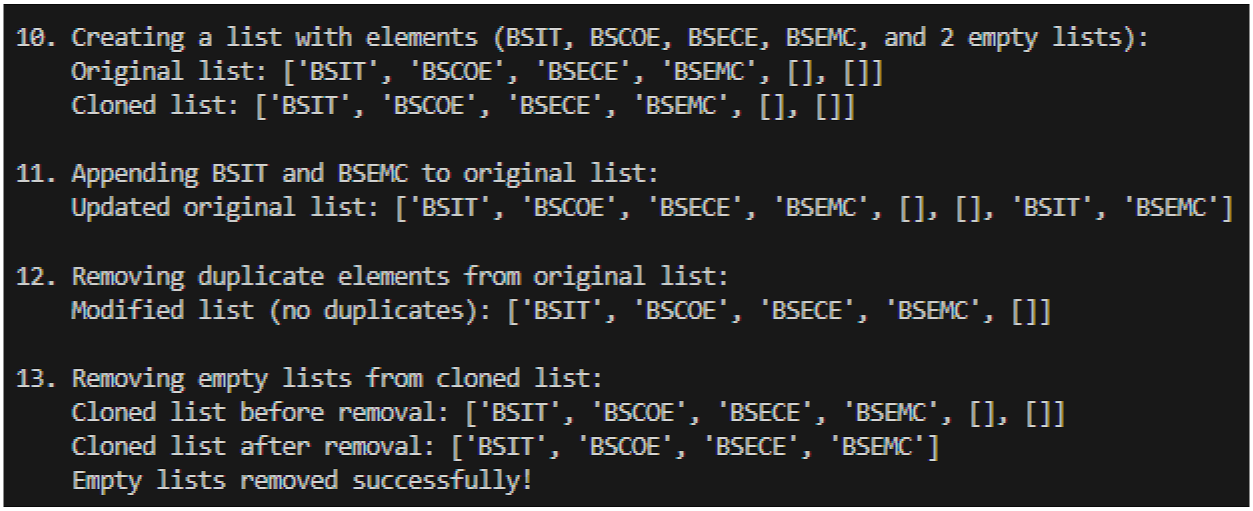
12. Remove the duplicate elements of the updated original list(from instruction no.12). Print the modified list.

13. Update your Cloned list by removing the empty list. Then print the updated cloned list



```python
# 10. Create a list and clone it
print("10. Creating a list with elements (BSIT, BSCOE, BSECE, BSEMC, and 2 empty lists):")
original_list = ['BSIT', 'BSCOE', 'BSECE', 'BSEMC', [], []]
cloned_list = original_list.copy()
print(f"    Original list: {original_list}")
print(f"    Cloned list: {cloned_list}")

# 11. Append 2 elements to original list
print("\n11. Appending BSIT and BSEMC to original list:")
original_list.append('BSIT')
original_list.append('BSEMC')
print(f"    Updated original list: {original_list}")

# 12. Remove duplicate elements
print("\n12. Removing duplicate elements from original list:")
# Preserve order while removing duplicates
seen = []
unique_list = []
for item in original_list:
    # Handle empty lists specially
    if item == []:
        if [] not in seen:
            seen.append([])
            unique_list.append(item)
    elif item not in seen:
        seen.append(item)
        unique_list.append(item)
original_list = unique_list
print(f"    Modified list (no duplicates): {original_list}")

print("\n13. Removing empty lists from cloned list:")
print(f"    Cloned list before removal: {cloned_list}")
cloned_list = [item for item in cloned_list if item != []]
print(f"    Cloned list after removal: {cloned_list}")
print(f"    Empty lists removed successfully!")
```

BERNARDO, Luke Joaquin - 302I



```
10. Creating a list with elements (BSIT, BSCOE, BSECE, BSEMC, and 2 empty lists):
    Original list: ['BSIT', 'BSCOE', 'BSECE', 'BSEMC', [], []]
    Cloned list: ['BSIT', 'BSCOE', 'BSECE', 'BSEMC', [], []]

11. Appending BSIT and BSEMC to original list:
    Updated original list: ['BSIT', 'BSCOE', 'BSECE', 'BSEMC', [], [], 'BSIT', 'BSEMC']

12. Removing duplicate elements from original list:
    Modified list (no duplicates): ['BSIT', 'BSCOE', 'BSECE', 'BSEMC', []]

13. Removing empty lists from cloned list:
    Cloned list before removal: ['BSIT', 'BSCOE', 'BSECE', 'BSEMC', [], []]
    Cloned list after removal: ['BSIT', 'BSCOE', 'BSECE', 'BSEMC']
    Empty lists removed successfully!
```