LAB CYCLE – 1

 # 1. Assign variables with various values of different data types, print the value and type of the variable

 # a. Numeric

 # i. Int

```
int_var = 10
print(f"Integer: {int_var}, Type: {type(int_var)}")
```

Output

Integer: 10, Type: <class 'int'>

 # ii. Float

```
float_var = 10.5
print(f"Float: {float_var}, Type: {type(float_var)}")
```

Output

Float: 10.5, Type: <class 'float'>

 # iii. Complex

```
complex_var = 3 + 5j
print(f"Complex: {complex_var}, Type: {type(complex_var)}")
```

Output

Complex: (3+5j), Type: <class 'complex'>

 # b. String

```
string_var = "Hello World"
# i. Print complete string
print(f"Complete String: {string_var}")
```

Output

Complete String: Hello World

```
# ii. Print first character of the string
print(f"First Character: {string_var[0]}")
```

Output

First Character: H

```
# iii. Print characters starting from 3rd to 5th
print(f"3rd to 5th Characters: {string_var[2:5]}")
```

Output

3rd to 5th Character: llo

```
# iv. Print string starting from 3rd character
print(f"String from 3rd Character: {string_var[2:]}")
```

Output

```
String from 3rd Character: llo World
```

```
# v. Print string two times
print(f"String Two Times: {string_var * 2}")
```

Output

```
String Two Times: Hello WorldHello World
```

```
# vi. Print concatenated string
concatenated_str = string_var + " Python"
print(f"Concatenated String: {concatenated_str}")
```

Output

```
Concatenated String: Hello World Python
```

 # c. Boolean

```
bool_var = True
print(f"Boolean: {bool_var}, Type: {type(bool_var)}")
```

Output

```
Boolean: True, Type: <class 'bool'>
```

 # d. List

```
list_var = [1, 2, 3, 4, 5]
# i. Print complete list
print(f"Complete List: {list_var}")
```

Output

```
Complete List: [1, 2, 3, 4, 5]
```

```
# ii. Print first element of the list
print(f"First Element: {list_var[0]}")
```

Output

```
First Element: 1
```

```
# iii. Print elements starting from 2nd till 3rd
print(f"2nd to 3rd Elements: {list_var[1:3]}")
```

Output

2nd to 3rd Element: [2, 3]

iv. Print elements starting from 3rd element

print(f"Elements from 3rd Element: {list_var[2:]}")

Output

Elements from 3rd: [3, 4, 5]

v. Print list two times

print(f"List Two Times: {list_var * 2}")

Output

List Two Times: [1, 2, 3, 4, 5, 1, 2, 3, 4, 5]

vi. Print concatenated lists

concatenated_list = list_var + [6, 7]

print(f"Concatenated List: {concatenated_list}")

Output

Concatenated List: [1, 2, 3, 4, 5, 6, 7]

vii. Find the number of elements in a list

print(f"Number of Elements in List: {len(list_var)}")

Output

Length of List: 5

 # e. Tuple

tuple_var = (10, 20, 30, 40, 50)

i. Print the complete tuple

print(f"Complete Tuple: {tuple_var}")

Output

Complete Tuple: (10, 20, 30, 40, 50)

ii. Print first element of the tuple

print(f"First Element of Tuple: {tuple_var[0]}")

Output

First Element: 10

iii. Print elements of the tuple starting from 2nd till 3rd

print(f"2nd to 3rd Elements of Tuple: {tuple_var[1:3]}")

Output

2nd to 3rd Element: (20, 30)

iv. Print elements of the tuple starting from 3rd element

print(f"Elements from 3rd Element: {tuple_var[2:]}")

Output

Elements from 3rd: (30, 40, 50)

v. Print the contents of the tuple twice

print(f"Tuple Two Times: {tuple_var * 2}")

Output

Tuple Twice: (10, 20, 30, 40, 50, 10, 20, 30, 40, 50)

vi. Print concatenated tuples

concatenated_tuple = tuple_var + (60, 70)

print(f"Concatenated Tuple: {concatenated_tuple}")

Output

Concatenated Tuple: (10, 20, 30, 40, 50, 60, 70)

vii. Find the number of elements in a tuple

print(f"Number of Elements in Tuple: {len(tuple_var)}")

Output

Length of Tuple: 5

 # f. Dictionary

dict_var = {'name': 'Alice', 'age': 25, 'city': 'New York'}

i. Print value for a particular key

print(f"Value for 'name': {dict_var['name']}")

Output

Value for 'name': Alice

ii. Print value for 2nd key

print(f"Value for 'age': {dict_var['age']}")

Output

Value for 'age': 25

iii. Print complete dictionary

```
print(f"Complete Dictionary: {dict_var}")
```

Output

```
Complete Dictionary: {'name': 'Alice', 'age': 25, 'city': 'New York'}
```

```
# iv. Print all the keys
print(f"All Keys: {dict_var.keys()}")
```

Output

```
Keys: dict_keys(['name', 'age', 'city'])
```

```
# v. Print all the values
print(f"All Values: {dict_var.values()}")
```

Output

```
Values: dict_values(['Alice', 25, 'New York'])
```

 # g. Set

```
set_var = {1, 2, 3, 4, 5}
# i. Print complete set
print(f"Complete Set: {set_var}")
```

Output

```
Complete Set: {1, 2, 3, 4, 5}
```

```
# ii. Find the number of elements in a set
print(f"Number of Elements in Set: {len(set_var)}")
```

Output

```
Length of Set: 5
```

```
# iii. Add an item to a set, using the add() method
set_var.add(6)
print(f"Set After Adding 6: {set_var}")
```

Output

```
Set After Adding 6: {1, 2, 3, 4, 5, 6}
```

```
# iv. Add items from another set into the current set, using update() method
set_var.update({7, 8})
print(f"Set After Update: {set_var}")
```

Output

```
Set After Update: {1, 2, 3, 4, 5, 6, 7, 8}
```

```
# v. Remove an item in a set, using the remove() method
set_var.remove(8)
print(f"Set After Removing 8: {set_var}")
```

Output

```
Set After Removing 8: {1, 2, 3, 4, 5, 6, 7}
```

```
# vi. Remove an item in a set, using the pop() method
set_var.pop()
print(f"Set After Pop: {set_var}")
```

Output

```
Set After Pop: {2, 3, 4, 5, 6, 7}
```

```
# vii. Remove all items in a set using clear() method
set_var.clear()
print(f"Set After Clear: {set_var}")
```

Output

```
Set After Clear: set()
```

```
# viii. Delete the set using del() method
del set_var
```

Output

```
# Trying to print will raise an error as the set no longer exists.
```

 # 2. Perform string operations

```
# a. Concatenate two strings with space in between
str1 = "Hello"
str2 = "World"
concatenated_with_space = str1 + " " + str2
print(f"Concatenated with space: {concatenated_with_space}")
```

Output

```
Concatenated with space: Hello World
```

```
# b. Create a new string by formatting it using the placeholders and values
name = "Alice"
age = 25
formatted_string = f"My name is {name} and I am {age} years old."
print(f"Formatted String: {formatted_string}")
```

Output

Formatted String: My name is Alice and I am 25 years old.

```
# c. Find the length of a string
length_of_string = len(str1)
print(f"Length of string: {length_of_string}")
```

Output

Length of string: 5

```
# d. Convert to upper case
upper_case_string = str1.upper()
print(f"Uppercase: {upper_case_string}")
```

Output

Uppercase: HELLO

```
# e. Convert to lower case
lower_case_string = str2.lower()
print(f"Lowercase: {lower_case_string}")
```

Output

Lowercase: world

```
# f. Converts the first character to upper case
capitalized_string = str2.capitalize()
print(f"Capitalized: {capitalized_string}")
```

Output

Capitalized: World

```
# g. Access individual characters of a string by indexing
print(f"First character of str1: {str1[0]}")
```

Output

First character of str1: H

```
# h. Splits the string at the specified separator and returns a list
split_string = concatenated_with_space.split(" ")
print(f"Splitted String: {split_string}")
```

Output

Split String: ['Hello', 'World']

```
# i. Replace a string with another string
replaced_string = concatenated_with_space.replace("World", "Python")
print(f"Replaced String: {replaced_string}")
```

Output

```
Replaced String: Hello Python
```

```
# j. Finds the position of a substring within a string
substring_position = concatenated_with_space.find("World")
print(f"Position of 'World': {substring_position}")
```

Output

```
Position of 'World': 6
```

```
# k. Removes leading and trailing spaces
whitespace_string = "   Hello Python   "
trimmed_string = whitespace_string.strip()
print(f"Trimmed String: '{trimmed_string}'")
```

Output

```
Trimmed String: 'Hello Python'
```

 # 3. Range function

```
# a. Create a list containing the numbers from 0 up to, but not including, 10
range_0_to_9 = list(range(10))
print(f"Range 0 to 9: {range_0_to_9}")
```

Output

```
Range from 0 to 9: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
# b. Create a list containing the numbers from 1 up to, but not including, 10
range_1_to_9 = list(range(1, 10))
print(f"Range 1 to 9: {range_1_to_9}")
```

Output

```
Range from 1 to 9: [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
# c. Create a list containing a sequence of numbers starting from 1 (inclusive), up to, but not including, 10, with a step of 2
range_step_2 = list(range(1, 10, 2))
print(f"Range from 1 to 9 with step 2: {range_step_2}")
```

Output

```
Range from 1 to 9 with step 2: [1, 3, 5, 7, 9]
```

 # 4. Containers - List operations

```
list_example = [10, 20, 30, 40, 50]
# a. Append
list_example.append(60)
print(f"List after append: {list_example}")
```

Output

List after append: [10, 20, 30, 40, 50, 60]

```
# b. Pop
popped_element = list_example.pop()
print(f"Popped element: {popped_element}, List after pop: {list_example}")
```

Output

Popped element: 60, List after pop: [10, 20, 30, 40, 50]

```
# c. Insert an element at a specific index within a list
list_example.insert(2, 25)
print(f"List after inserting 25 at index 2: {list_example}")
```

Output

List after inserting 25 at index 2: [10, 20, 25, 30, 40, 50]

```
# d. Remove the first occurrence of a specified element from a list
list_example.remove(25)
print(f"List after removing 25: {list_example}")
```

Output

List after removing 25: [10, 20, 30, 40, 50]

```
# e. Append list to another list
list_to_append = [70, 80]
list_example.append(list_to_append)
print(f"List after appending another list: {list_example}")
```

Output

List after appending another list: [10, 20, 30, 40, 50, [70, 80]]

```
# f. Remove appended list
list_example.remove(list_to_append)
print(f"List after removing appended list: {list_example}")
```

Output

List after removing appended list: [10, 20, 30, 40, 50]

g. Extend the list with another list

list_example.extend(list_to_append)

print(f"List after extending with another list: {list_example}")

Output

List after extending with another list: [10, 20, 30, 40, 50, 70, 80]

h. Sort a list

list_example.sort()

print(f"Sorted List: {list_example}")

Output

Sorted List: [10, 20, 30, 40, 50, 70, 80]

 # 5. List Slicing

numbers = list(range(11))

a. Print values of second and last element

print(f"Second element: {numbers[1]}, Last element: {numbers[-1]}")

Output

Second element: 1, Last element: 10

b. Slice elements from index 5 (inclusive) to index 11 (exclusive)

sliced_list = numbers[5:11]

print(f"Sliced List (5 to 11): {sliced_list}")

Output

Sliced List (5 to 11): [5, 6, 7, 8, 9, 10]

c. Slice elements from index 5 (inclusive) to the end of the list

slice1 = numbers[5:]

print(f"Sliced List from index 5: {slice1}")

Output

Sliced List from index 5: [5, 6, 7, 8, 9, 10]

d. Slice elements from the beginning to index 7 (exclusive)

slice2 = numbers[:7]

print(f"Sliced List (0 to 7): {slice2}")

Output

Sliced List (0 to 7): [0, 1, 2, 3, 4, 5, 6]

e. Slice elements from the second-to-last element to the end

slice3 = numbers[-2:]

print(f"Sliced List from second-to-last element: {slice3}")

Output

Sliced List from second-to-last element: [9, 10]