ABBOTTABAD UNIVERSITY SCIENCE AND TECHENOLOGY

LAB TASK NO 2&3

NAME ***MUHAMMAD TAYYAB RASHEED***

ROLL NO ***F24-634***

LAB TASK NO ***2&3***

SUBMITTED TO ***SIR JAMAL ABDUL AHAD***

SUBJECT ***OBJECT ORENTED PROGRAMMING***

PROGRAM ***BS SOFTWARE ENGINEERING***

SEMESTER ***SECOND SEMESTER***

SECTION ***C***

```python
#Excrise no 1
def hello_name(name):
    print("Hello, " + name + "!")

# Example usage:
hello_name("Ali")
```

```
Hello, Ali!
```

```python
#Excrise no 2
def calculate_area(length, width=10):
    return length * width

# Example usage:
print(calculate_area(5))  # Output: 50 (using default width of 10)
print(calculate_area(5, 20))  # Output: 100 (using custom width of 20)
```

```
====================
50
100
>
```

```python
#Excirse no 3

def is_even(num):
    return num % 2 == 0

# Example usage:
print(is_even(10))  # Output: True
print(is_even(11))  # Output: False
```

```
= RESTART: C:/User
True
False
```

```python
#Excirse no 4
x = 10  # global variable

def modify_global():
    global x  # declare x as global
    x = 20  # modify the global variable
    print("Inside function:", x)

modify_global()
print("Outside function:", x)
```

```
= RESTART: C:/Users/Global/
Inside function: 20
Outside function: 20
```

```python
#Excrise no 7:
def add(a, b):
    return a + b

def subtract(a, b):
    return a - b

def multiply(a, b):
    return a * b

def operate(func, a, b):
    return func(a, b)

# Example usage:
print(operate(add, 10, 5))  # Output: 15
print(operate(subtract, 10, 5))  # Output: 5
print(operate)
```

```
15
5
<function operate at 0x0000026CB36CDE40>
```

LAB TASK NO 3:

```
#List Operations
# Define two lists
list1 = [1, 2, 3]
list2 = [4, 5, 6]
# 1. Combine lists
combined = list1 + list2
print("Combined List:", combined)
# 2. Repeat elements
repeated = list1 * 3
print("Repeated List:", repeated)
# 3. Check membership
print("2 in list1:", 2 in list1)  # True
print("7 not in list2:", 7 not in list2)  # True
```

```
= RESTART: C:/Users/Global/AppData/Local/Programs/Pyth
Combined List: [1, 2, 3, 4, 5, 6]
Repeated List: [1, 2, 3, 1, 2, 3, 1, 2, 3]
2 in list1: True
7 not in list2: True
```

```python
#TASK NO 2:
#List Operations:

# Define a list
numbers = [1, 2, 3]

# 1. Append and extend
numbers.append(4)
numbers.extend([5, 6])
print("Appended and Extended List:", numbers)

# 2. Insert, remove, and pop
numbers.insert(2, 99)  # Insert 99 at index 2
print("List after insertion:", numbers)

numbers.remove(99)  # Remove first occurrence
print("List after removal:", numbers)

last_item = numbers.pop()  # Remove last item
print("Last item popped:", last_item)
print("List after popping:", numbers)
```

```
Appended and Extended List: [1, 2, 3, 4, 5, 6]
List after insertion: [1, 2, 99, 3, 4, 5, 6]
List after removal: [1, 2, 3, 4, 5, 6]
Last item popped: 6
List after popping: [1, 2, 3, 4, 5]
>
```

```python
# TASK NO 3:
# Loop Iterations:
# Define a list
fruits = ["apple", "banana", "cherry"]
# 1. For loop iteration
print("For Loop Iteration:")
for fruit in fruits:
    print(fruit)
# 2. While loop iteration
print("\nWhile Loop Iteration:")
i = 0
while i < len(fruits):
    print(fruits[i])
    i += 1
# 3. Using enumerate for index tracking
print("\nUsing Enumerate for Index Tracking:")
for index, fruit in enumerate(fruits):
    print(index, fruit)
```

```
For Loop Iteration:
apple
banana
cherry

While Loop Iteration:
apple
banana
cherry

Using Enumerate for Index Tracking:
0 apple
1 banana
2 cherry
```

File Edit Format Run Options Window Help

```
#TASK NO 4:
#List Comprehensions
 #1 Basic list comprehension
print("Basic List Comprehension:")
squares = [x**2 for x in range(10)]
print(squares)
  #2 Conditional list comprehension
print("\nConditional List Comprehension:")
even_numbers = [x for x in range(10) if x % 2 == 0]
print(even_numbers)
  #3 Nested list comprehensions
print("\nNested List Comprehensions:")
matrix = [[i * j for j in range(3)] for i in range(3)]
for row in matrix:
    print(row)
```

```
Basic List Comprehension:
[0, 1, 4, 9, 16, 25, 36, 49, 64, 81]

Conditional List Comprehension:
[0, 2, 4, 6, 8]

Nested List Comprehensions:
[0, 0, 0]
[0, 1, 2]
[0, 2, 4]
```

```python
# TASK NO 5:
#Deque Operations and List Iteration

from collections import deque

# Deque operations
print("Deque Operations:")
d = deque([1, 2, 3])
d.appendleft(0)
print(d)

# Avoid modifying lists during iteration
print("\nAvoid Modifying Lists During Iteration:")
numbers = [1, 2, 3, 4, 5]
numbers = [x for x in numbers if x % 2 == 0]  # Remove odd numbers
print(numbers)
```

```
= RESTART: C:/Users/Global/AppData/Local/Programs/Py
Deque Operations:
deque([0, 1, 2, 3])

Avoid Modifying Lists During Iteration:
[2, 4]
```

```python
# TASK NO 6:
#Statistics and To-Do List Manager
# Statistics
print("Statistics:")
marks = [88, 76, 90, 85, 72]
highest = max(marks)
lowest = min(marks)
average = sum(marks) / len(marks)
print(f"Highest: {highest}, Lowest: {lowest}, Average: {average:.2f}")
# To-Do List Manager
print("\nTo-Do List Manager:")
tasks = []
while True:
    print("Options:")
    print("1. Add task")
    print("2. Remove task")
    print("3. Display tasks")
    print("4. Quit")
    choice = input("Choose an option: ")
    if choice == "1":
        task = input("Enter a task: ")
        tasks.append(task)
        print("Task added!")
    elif choice == "2":
        task = input("Enter a task to remove: ")
        if task in tasks:
            tasks.remove(task)
            print("Task removed!")
        else:
            print("Task not found!")
    elif choice == "3":
        print("Tasks:")
        for i, task in enumerate(tasks, start=1):
            print(f"{i}. {task}")
    elif choice == "4":
        print("Goodbye!")
        break
    else:
        print("Invalid option. Please choose again.")
```

```
Statistics:
Highest: 90, Lowest: 72, Average: 82.20

To-Do List Manager:
Options:
1. Add task
2. Remove task
3. Display tasks
4. Quit
Choose an option:
```