OBJECT ORIENTED PROGRAMMING

LAB TASK 3

NAME: MASHAR KHAN

ROLL NO: F24_623

SEMESTER: 2nd

SECTION: C

SUBJECT: Object Oriented Programming

SUBMITTED TO: MR Jamal Abdul Ahad

DEPARTMENT OF Computer science

CREATING AND ACCESSING LIST

QUESTION NO: 1

Create a list of 5 elements and print the second and last element?

Modify the third element and print the update list?

LAB 3.py - D:/SEMESTER 2/OOP,S ASSIGMENT/LAB 3.py (3.13.2)

File Edit Format Run Options Window Help

```
# Create a list with at least 5 elements
my_list = [10, 20, 30, 40, 50]

# Print the second and last element
print("Second element:", my_list[1])
print("Last element:", my_list[-1])

# Modify the third element
my_list[2] = 99

# Print the updated list
print("Updated list:", my_list)
```

IDLE Shell 3.13.2

File Edit Shell Debug Options Window Help

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/LAB 3.py ================
Second element: 20
Last element: 50
Updated list: [10, 20, 99, 40, 50]
>>>
```

LIST OPERATIONS

QUESTION NO : 2

Create two list merge them and check if a specific element exist in the merge list?

LAB 4.py - D:/SEMESTER 2/OOP,S ASSIGMENT/LAB 4.py (3.13.2)

File Edit Format Run Options Window Help

```
# Create two lists
list1 = [1, 2, 3, 4, 5]
list2 = [6, 7, 8, 9, 10]

# Merge the lists
merged_list = list1 + list2

# Check if a specific element exists in the merged list
element_to_check = 7
if element_to_check in merged_list:
    print(f"{element_to_check} exists in the merged list.")
else:
    print(f"{element_to_check} does not exist in the merged list.")

# Print the merged list
print("Merged list:", merged_list)
```

IDLE Shell 3.13.2

File Edit Shell Debug Options Window Help

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/LAB 4.py ==========
7 exists in the merged list.
Merged list: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
>>>
```

COMMAN LIST METHOD

QUESTION NO:3

Create a list and perform append, extend, insert, remove, pop, sort, and reverse operations?

JDF U.py - D:/SEMESTER 2/OOP,S ASSIGMENT/JDF U.py (3.13.2)

File Edit Format Run Options Window Help

```
# Create a list
my_list = [10, 20, 30, 40, 50]

# Append an element
my_list.append(60)
print("After append:", my_list)

# Extend the list with another list
my_list.extend([70, 80, 90])
print("After extend:", my_list)

# Insert an element at a specific position (index 2)
my_list.insert(2, 25)
print("After insert at index 2:", my_list)

# Remove an element (first occurrence of 30)
my_list.remove(30)
print("After removing 30:", my_list)

# Pop the last element
popped_element = my_list.pop()
print("Popped element:", popped_element)
print("After pop:", my_list)

# Sort the list in ascending order
my_list.sort()
print("After sort:", my_list)

# Reverse the list
my_list.reverse()
print("After reverse:", my_list)
```

IDLE Shell 3.13.2

File Edit Shell Debug Options Window Help

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/JDF U.py ==========
After append: [10, 20, 30, 40, 50, 60]
After extend: [10, 20, 30, 40, 50, 60, 70, 80, 90]
After insert at index 2: [10, 20, 25, 30, 40, 50, 60, 70, 80, 90]
After removing 30: [10, 20, 25, 40, 50, 60, 70, 80, 90]
Popped element: 90
After pop: [10, 20, 25, 40, 50, 60, 70, 80]
After sort: [10, 20, 25, 40, 50, 60, 70, 80]
After reverse: [80, 70, 60, 50, 40, 25, 20, 10]
>>>
```

Iterating over list

QUESTION NO : 4

Write a program to print each element of a list with in index using enumerate?

```
e67t67et7te.py - D:/SEMESTER 2/OOP,S ASSIGMENT/e67t67et7te.py (3.13.2)
File  Edit  Format  Run  Options  Window  Help
# Sample list
my_list = ['apple', 'banana', 'cherry', 'date']

# Using enumerate to print index and element
for index, value in enumerate(my_list):
    print(f"Index {index}: {value}")
```

```
IDLE Shell 3.13.2
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/e67t67et7te.py ==========
Index 0: apple
Index 1: banana
Index 2: cherry
Index 3: date
>>>
```

LIST COMPREHENSIONS

QUESTION NO: 5

Create a list of numbers from 1 to 20 containing only multiples of 3 using list comprehension?

```
zxcvb.py - D:/SEMESTER 2/OOP,S ASSIGMENT/zxcvb.py (3.13.2)
File  Edit  Format  Run  Options  Window  Help
# List comprehension to get multiples of 3 from 1 to 20
multiples_of_3 = [num for num in range(1, 21) if num % 3 == 0]

# Print the result
print(multiples_of_3)
```

```
IDLE Shell 3.13.2
File  Edit  Shell  Debug  Options  Window  Help
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/zxcvb.py ================
[3, 6, 9, 12, 15, 18]
>>>
```

WORKING WITH NESTED LISTS

QUESTION NO: 6

Write a program to find the sum of all elements in a nested list?

sfdfg344.py - D:/SEMESTER 2/OOP,S ASSIGMENT/sfdfg344.py (3.13.2)

File Edit Format Run Options Window Help

```
# Function to find the sum of all elements in a nested list
def nested_list_sum(lst):
    total = 0
    for item in lst:
        if isinstance(item, list):
            total += nested_list_sum(item)
        else:
            total += item
    return total
nested_list = [11, [72, 83], [64, [95, 66]], 97]

result = nested_list_sum(nested_list)

# Print result
print("Sum of all elements:", result)
```

IDLE Shell 3.13.2

File Edit Shell Debug Options Window Help

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/sfdfg344.py ==============
Sum of all elements: 488
>>>
```

LIST PERFORMANCE AND OPTIMAZATION

QUESTION NO: 7

Use deque to implement a queue where elements are added and removed efficiently?

789543.py - D:/SEMESTER 2/OOP,S ASSIGMENT/789543.py (3.13.2)

File Edit Format Run Options Window Help

```
from collections import deque

# Initialize a deque as a queue
queue = deque()

# Enqueue (Add elements to the queue)
queue.append("Alice")
queue.append("Bob")
queue.append("Charlie")

print("Queue after enqueuing:", queue)

# Dequeue (Remove elements from the front of the queue)
first = queue.popleft()
print("Dequeued:", first)

second = queue.popleft()
print("Dequeued:", second)

print("Queue after dequeuing:", queue)
```

IDLE Shell 3.13.2

File Edit Shell Debug Options Window Help

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
============== RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/789543.py ==============
Queue after enqueuing: deque(['Alice', 'Bob', 'Charlie'])
Dequeued: Alice
Dequeued: Bob
Queue after dequeuing: deque(['Charlie'])
>>>
```

REAL WORLD APPLICATOION OF LISTS

QUESTION NO: 8

Build a Python program that manages a list of tasks (adding, removing, displaying) with a menu-driven interface?

ertyu.py - D:/SEMESTER 2/OOP,S ASSIGMENT/ertyu.py (3.13.2)

File Edit Format Run Options Window Help

```
def display_tasks(tasks):
    """Displays the current task list."""
    if not tasks:
        print("\nNo tasks available.")
    else:
        print("\nCurrent Tasks:")
        for index, task in enumerate(tasks, start=1):
            print(f"{index}. {task}")
def add_task(tasks):
    """Adds a task to the list."""
    task = input("Enter the task: ")
    tasks.append(task)
    print(f"Task '{task}' added successfully!")
def remove_task(tasks):
    """Removes a task by index."""
    display_tasks(tasks)
    try:
        index = int(input("\nEnter task number to remove: ")) - 1
        if 0 <= index < len(tasks):
            removed_task = tasks.pop(index)
            print(f"Task '{removed_task}' removed successfully!")
        else:
            print("Invalid task number!")
    except ValueError:
        print("Please enter a valid number!")
def main():
    """Main function to display menu and handle user input."""
    tasks = []  # List to store tasks
    while True:
        print("\nTask Manager Menu:")
        print("1. Add Task")
        print("2. Remove Task")
        print("3. Display Tasks")
        print("4. Exit")
        choice = input("Enter your choice: ")
        if choice == '1':
            add_task(tasks)
        elif choice == '2':
            remove_task(tasks)
        elif choice == '3':
            display_tasks(tasks)
        elif choice == '4':
            print("Exiting Task Manager. Goodbye!")
            break
        else:
            print("Invalid choice! Please select a valid option.")
```

IDLE Shell 3.13.2

File Edit Shell Debug Options Window Help

```
Python 3.13.2 (tags/v3.13.2:4f8bb39, Feb  4 2025, 15:23:48) [MSC v.1942 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
================ RESTART: D:/SEMESTER 2/OOP,S ASSIGMENT/ertyu.py ================

Task Manager Menu:
1. Add Task
2. Remove Task
3. Display Tasks
4. Exit
Enter your choice:
```