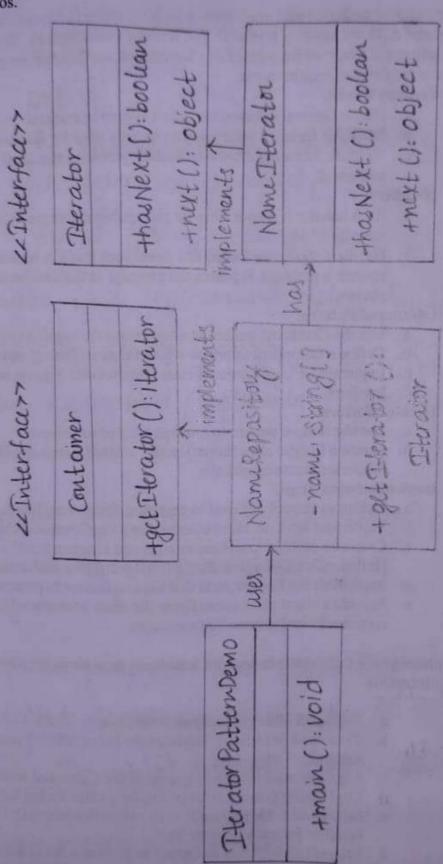
Functiment#	Student ID	
Experiment#	Student Name	

2) Draw the UML Relationship Diagram for Iterator Design Pattern for customized Scenarios.



Course Title	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
Course Code	23CS2103A & 23CS2103E	Page   67

Experiment# Student ID Student Name

at processo he system sign patten

nosing probled to be

of a specific

ndler cannot nandler in the

requests. message) me ng requests u

ls. and objects an

nd class.
andler classes
th severity lest
process common onsibility, cre-

he above

els. ring message)

and interface r log handlers asses); Concre

ist of command he logging system public enum Loglevel [

INFO,

Procedure/Program:

DEBUG,

ERROR

3 public interface Command [

void execute (string message);

3 public class LogCommand implements Command [

private LogHandler handler;

public LogCommand (LogHandler handler) {

this handler = handler;

} public void execute (string message) [

handler. handle Request (message);

} public abstract class LogHandler [

protected logHandler nextHandler;

public void set Next Handler (Log Handler next Handler)

this n extHandler = nextHandler.

Course Title Advanced Object-Oriented Programming ACADEMIC YEAR: 2024-25
Course Code 23CS2103A & 23CS2103E Page | 69

DEMIC YEAR: N

	Student ID	
Experiment#	Student Name	
Date		

public abstract void handle Request (String message);
}

public class Info Handler extends LogHandler {

public void handleRequest (string message) {

if (message contains (loglevel INFO name ())) {

System out println ("INFO Handler: " + message);

} else if (next Handler != null) {

next Handler handle Request (message);

Jewelic class Debugtlandler extends Logtlandler public void handleRequest (string message) {

if (message contains (Loglevel DERUG name ())) {

system out println ("DEBUG Handler: " + message);

yelse if (next Handler != null) {

wext Handler (message).

Course Title	Advan
Course Code	Advanced Object-Or
	23CS2103A & 23CC

Experiment#	Student ID	
Date	Student Name	

3
3 public class Error Handler

public class Error Handler extends LogHandler [
public void handle Request (string message) [
if (message contains (LogLevel ERROR name (1))) [

system-out-println ("ERROR Handler:" + message);

Jelse if (next Handler != null) [

nextHandler. handle Request (message).

3 3

import java. util. Arraylist;

import java util Iterator;

import java-util· List;

public class Logger (

private List & Command 7 commands = new

Arraylister ();

public void add Command (Command command) {
commands add (Command);

} public void process Commands () {

Course Title Advanced Object-Oriented Programming ACADEMIC YEAR: 2024-25
Course Code 23CS2103A & 23CS2103E Page | 71

	Student ID	
1	Student Name	
Experiment#		
Date		

Iterator & Command iterator = commands · Herator().

While (iterator · hasNext (1)) {

Command command = iterator · next ();

command · executi ("log message of severity level,

INFO, DEBUG or ERROR");

3

3

Course Title Advanced Object-Oriented Programming ACADEMIC V

Experiment#	Student ID
Date	Student Name

### ✓ Data and Results:

INFO, DEBUG or ERROR

DEBUG Handler: Log message of severity level

INFO, DEBUG or ERROR

ERROR Handler: Log message of severity level

INFO, DEBUG or ERROR

INFO, DEBUG or ERROR

### ✓ Analysis and Inferences:

The integration of the Chain of Responsibility, command and Iterator patterns in the logging system ensures that it is well-organized, flexible and easy to maintain, allowing for effective handling of various log levels and logging operations.

Course Title	Advanced Object O. J. C. L.	
The state of the s	Advanced Object-Oriented Programming	ACADEMIC YEAR: 2024-25
course code	23CS2103A & 23CS2103E	Page   75

	Student ID	-
Experiment#	Student Name	-
Date		

# VIVA-VOCE Questions (In-Lab):

1) State at which situation that we need Chain of Responsibility Design Patts The Chain of Responsibility design pattern is particularly useful in situations where you need to process a request through a series of handlers, each of which may or may not handle the request-

Discuss the Pros and Cons of Iterator Design Pattern.

Iterator Design pattern provides valuable abstraction and flexibility for traversing collections, but it may introduce additional complexity and overhead in certain scenarios.

3) Discuss the Pros and Cons of Command Design Pattern Command Design Pattern provides significant advantages in terms of flexibility, decoupling and support for complex operations such as undo-It may introduce additional complexity and overhead, particularly for simple requests.

Course Title Course Code	23CS2103A & 23CS2103E	1
Cour	2032T03E	ACADEMIC
		Page   76

4) Which pattern is used to hide the implementation details while traversing over an ArrayList?  A: The Iterator Pattern is used to hide implementation details while traversing an ArrayList. It allows sequential access to elements without exposing the collection's internal structure.

Experiment# Student ID

Date

Student Name

## ✓ Data and Results:

INFO Handler: Log message of severity level
INFO, DEBUG or ERROR

DEBUG Handler: Log message of severity level

INFO DEBUG OF ERROR

ERROR Handler: Log message of severity level

INFO DEBUG OF ERROR

### ✓ Analysis and Inferences:

The integration of the chain of responsibility, command and Iterator patterns in the logging system ensures that it is well-organized, flexible and easy to maintain.

Evaluator Remark (if Any):	
	Marks Secured:out of 50
Evaluator MUST and American	Signature of the Evaluator with Date
Course Title Advanced Object-Orio	r to signing and posting marks for each experiment