Class Exercise:

Decorators v1

You are tasked with creating a program that will process a list of students' grades and perform various operations on it using lambda functions, the filter() function, the map() function, and decorators. The program should have the following functionality:

1. Create a list of grades for a group of students. Each grade is represented as a number between 0 and 100, inclusive.
2. Use the filter() function with a lambda function to filter out all the grades that are below 60. Store the filtered grades in a new list.
3. Use the map() function with a lambda function to increase all the grades in the filtered list by 5 points. Store the modified grades in another new list.
4. Create a decorator function that will calculate and print the average grade of the modified grades list. The decorator should print the average grade after executing the decorated function.

Write the program to implement the above functionality and test it with a sample list of grades.

# Decorator function to calculate and print the average grade

def average\_grade\_decorator(func):

    def wrapper(grades):

# Your code here

    return wrapper

# Function to process the list of grades

@average\_grade\_decorator

def process\_grades(grades):

# Your code here

# Sample list of grades

grades\_list = [75, 80, 55, 90, 68, 72]

# Process the grades and get the modified grades list

modified\_grades\_list = process\_grades(grades\_list)

print("Modified Grades List:", modified\_grades\_list)

Expected output:

Average Grade: 80.0

Modified Grades List: [85, 90, 73, 95, 78, 82]