



UNIVERSITY OF CALOOCAN CITY  
COMPUTER ENGINEERING DEPARTMENT



Data Structure and Algorithm

Laboratory Activity No. 3

---

# Translating Algorithm to Program

---

*Submitted by:*  
Talagtag Mark Angel T.

*Instructor:*  
Engr. Maria Rizette H. Sayo

August 2, 2025

# I. Objectives

## Introduction

Data structure is a systematic way of organizing and accessing data, and an algorithm is a step-by-step procedure for performing some tasks in a finite amount of time. These concepts are central to computing, but to be able to classify some data structures and algorithms as “good,” we must have precise ways of analyzing them.

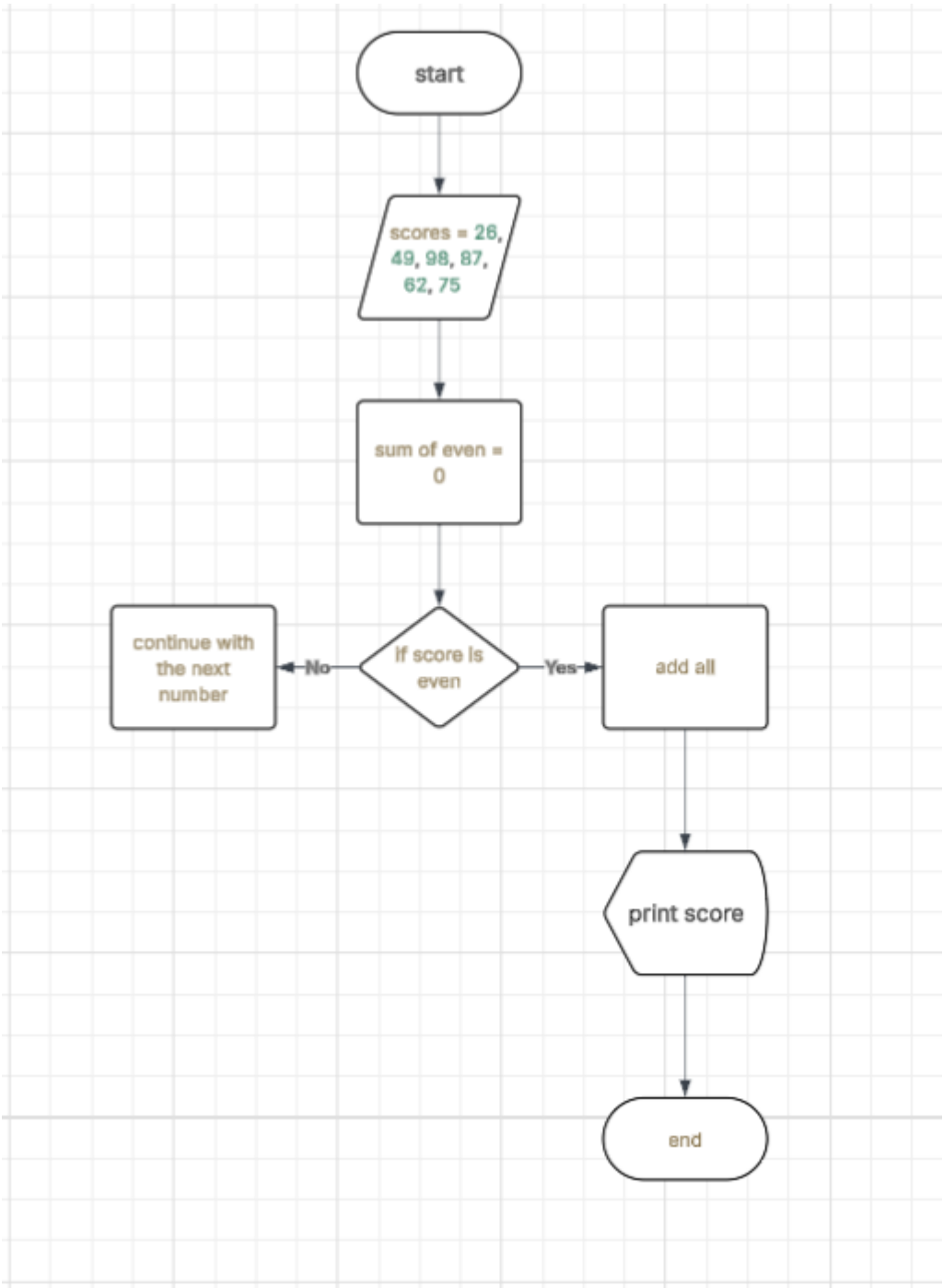
This laboratory activity aims to implement the principles and techniques in:

- Writing a well-structured procedure in programming
- Writing algorithm that best suits to solve computing problems
- Writing an efficient Python program from translated algorithms

# II. Methods

- Design an algorithm and the corresponding flowchart (Note: You may use LucidChart or any application) for adding the test scores as given below if the number is even: 26,49,98,87,62,75
- Translate the algorithm to a Python program (using Google Colab)
- Save your source codes to GitHub

III. Results



```
In [15]: scores = [26, 49, 98, 87, 62, 75]
sum_of_evens = 0
print("Starting the process...")

for score in scores:

    if score % 2 == 0:
        sum_of_evens += score

print("Process complete. Proceeding to output result...")
print("Sum of even test scores:", sum_of_evens)
temp_result = sum_of_evens + extra_variable
print("End of process.")

Starting the process...
Process complete. Proceeding to output result...
Sum of even test scores: 186
End of process.
```

Figure 1 Screenshot of program

If an image is taken from another literature or intellectual property, please cite them accordingly in the caption. Always keep in mind the Honor Code [1] of our course to prevent failure due to academic dishonesty.

## IV. Conclusion

This laboratory exercise serves as a valuable reminder of the essential relationship between code and flowcharts in software development. It highlights the critical importance of ensuring that code aligns with the flowchart, as they must work together cohesively to promote a deeper understanding of the program's logic and functionality. The flowchart, often considered a blueprint of the program, outlines the logical flow of the system in a visual form. By following this visual map when writing the code, developers can ensure that the program behaves as intended while maintaining clarity and precision throughout the development process.

The alignment between a flowchart and the corresponding code is fundamental for several reasons. First and foremost, it helps clarify the program's logic. When a flowchart is designed correctly, it breaks down the complex processes into manageable, understandable steps. These steps represent the flow of data and the decisions that the program must make. When coding, developers can follow the flowchart as a roadmap, ensuring that each step is implemented correctly and logically. Without this alignment, there is a significant risk of the program not working as expected or missing key steps.

## References

- [1] Co Arthur O.. “University of Caloocan City Computer Engineering Department Honor Code,” UCC-CpE Departmental Policies, 2020.