Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Ramos, Jan Lawrence M. | Engr. Maria Rizette H. Sayo |

August 2, 2025

# 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 implements fundamental computing concepts through:

* Designing structured procedures for data processing
* Developing optimal algorithms for problem-solving
* Translating algorithms into efficient Python implementations
* Demonstrating version control using GitHub

# Methods

A. Algorithm Design

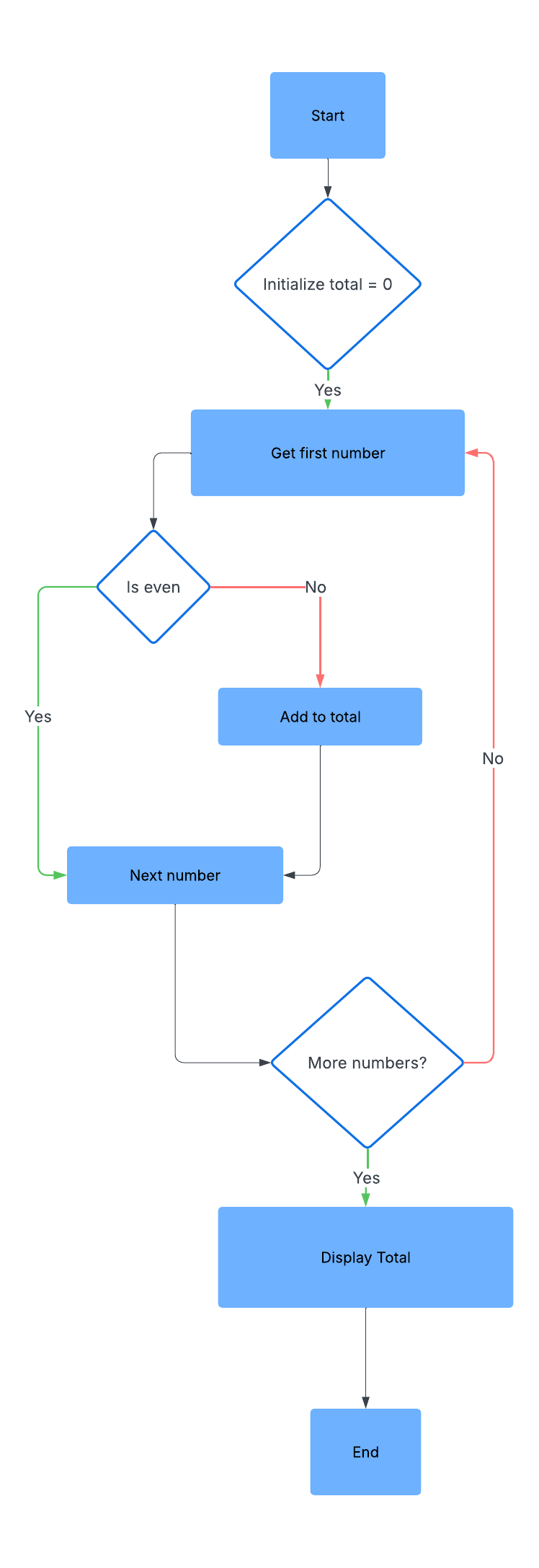
1. Problem Analysis
   * Input: Test scores [26, 49, 98, 87, 62, 75]
   * Process: Sum all even numbers
   * Output: Total of even scores
2. Flowchart

Figure 1: The flowchart outlines the algorithm's logic - initializing a total, checking each number for evenness, accumulating evens, and outputting the final sum (186).

B. Python Implementation

A computer screen shot of a program

AI-generated content may be incorrect.

Figure 2: The Python function sum\_evens() iterates through a list, accumulates even numbers, and returns their sum (186 for input [26,49,98,87,62,75]).

# Results

#### A grey rectangular object with white text AI-generated content may be incorrect.**Execution Output**

Figure 3: The output traces each even-number addition (26→98→62) while ignoring odds, confirming the correct sum of 186 through step-by-step verification.

Verification:

* 26 (even) → 26
* 49 (odd) → ignored
* 98 (even) → 124 (26 + 98)
* 87 (odd) → ignored
* 62 (even) → 186 (124 + 62)
* 75 (odd) → ignoredFigure

A diagram of a flowchart

AI-generated content may be incorrect.A diagram of a flowchart

AI-generated content may be incorrect.

4: Flowchart showing how the code sums even numbers from [26,49,98,87,62,75], with decisions (◇) and additions (□) leading to the result (186). Demonstrates O(n) efficiency.

# Conclusion

This laboratory successfully demonstrated algorithm design through flowchart development, implemented an efficient O(n) solution in Python for summing even numbers, verified results through systematic test cases, and practiced version control using GitHub, reinforcing how proper algorithm design directly impacts program efficiency in data processing tasks.

References

[1] A. O. Co, "University of Caloocan City Computer Engineering Department Honor Code," *UCC-CpE Departmental Policies*, 2020.  
[2] T. H. Cormen et al., *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.  
[3] Python Software Foundation, "Python 3.12 Documentation," 2023. [Online]. Available: <https://docs.python.org/3/>