



Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

Submitted by:

Ruperto, April Anne A.

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 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

ALGORITHM

1. Initialize a variable sum to 0.
2. Create a list of test scores: [26, 49, 98, 87, 62, 75]
3. For each number in the list:
 - If the number is even ($\text{number} \% 2 == 0$):
 - Add it to sum
4. After the loop, output the final value of sum.

FLOWCHART

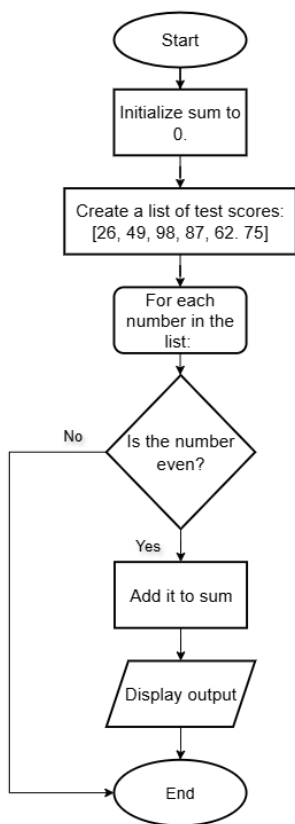


Figure 1: Flowchart of the Program

PYTHON PROGRAM

```
sum_even = 0
scores = [26, 49, 98, 87, 62, 75]

for score in scores:
    if score % 2 == 0:
        sum_even += score

print("Sum of even test scores:", sum_even)
```

Sum of even test scores: 186

Figure 2: Program

[DSA_Lab3.ipynb - Colab](#)

IV. Conclusion

This laboratory activity demonstrates the fundamental concepts of programming where a suitable algorithm was created, then it was transformed into a Python program and logical reasoning was used to solve a specific computer problem. Here, it was the addition of even numbers from the list of test scores. It also emphasizes the importance of creating a clear and logical approach, choosing the correct conditions (for example, checking if the numbers are even) and implementing them in the code. It illustrates how the algorithm acts as a roadmap to produce accurate, readable, and efficient Python software.

References

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