Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Hermosura, Leigh B. | Engr. Maria Rizette H. Sayo |

August 14, 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 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

# 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

# Results

This chapter showcases the outcome of this laboratory activity, accompanied by visual diagrams with their respective explanations.

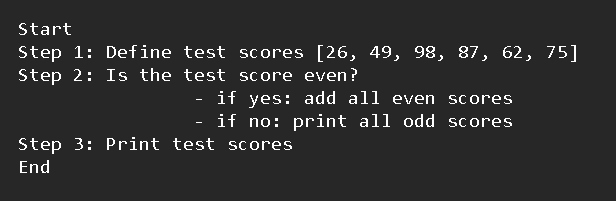


Figure 1 Algorithm of the program

The algorithm above presents the overall structure and process of the program. First, the test scores are defined, a function then checks the list of numbers if they’re even or odd. Once even numbers are determined, the program will sum up all even scores. If the scores are odd, they will be left as is. Finally, the program displays the test scores, the following is the flowchart of the algorithm.

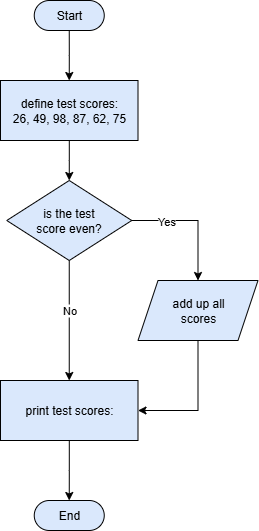


Figure 2 Flowchart of the program

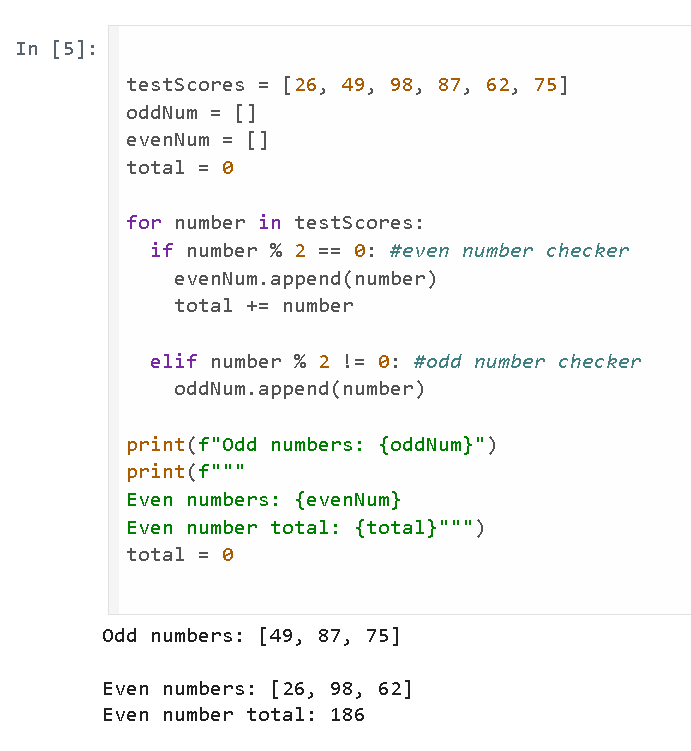
 Once the step-by-step procedure is clear, it is then translated into a Python program for automation.

Figure 3 Translation of the algorithm into code

The first part of the code defines the variables *testScores*, *oddNum*, *evenNum*, and *total*. The *testScores* variable is the provided set of scores that the program will check for even and odd numbers. The lists *oddNum* and *evenNum* are left empty as they will serve as the container for the separation of numbers.

A for loop then iterates over each items inside the *testScores* list and passes them through two conditions:

* If there are no remainders when the number is divided by 2, the number is even, appended to the list *evenNum*, and added to the *total* sum.
* If the number has a remainder when divided by 2, the number is odd and appended to the list *oddNum*.

After processing the rest of the numbers, the results are then printed and the *total* variable is reset to 0 to prevent accumulation of values when reused.

# Conclusion

This chapter demonstrates the process of algorithm development, from conceptualization to implementation. By visualizing the logic through flowcharts and then translating it into a Python program, the activity highlights how a clear step-by-step approach can lead to an efficient and organized solution.