Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Gabuyo, Ivan love D. | Engr. Maria Rizette H. Sayo |

August, 01, 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

Present the visualized procedures done. Also present the results with corresponding data visualizations such as graphs, charts, tables, or image . Please provide insights, commentaries, or explanations regarding the data. If an explanation requires the support of literature such as academic journals, books, magazines, reports, or web articles please cite and reference them using the IEEE format.

Please take note of the styles on the style ribbon as these would serve as the style format of this laboratory report. The body style is Times New Roman size 12, line spacing: 1.5. Body text should be in Justified alignment, while captions should be center-aligned. Images should be readable and include captions. Please refer to the sample below:

ALGORITHIM:  
1.Start  
2 Initialize a variable sum\_of\_evens to 0

3. For each score in the list:

* If score % 2 == 0 (i.e., the score is even):
  + Add score to sum\_of\_evens

4. After all scores have been checked, display sum\_of\_evens

5. **End**

**FLOWCHART:**

**1. Start**

* The beginning of the program.

**2. Initialize sum\_of\_evens to 0**

* This variable will hold the running total of even numbers.

**3. Input the list of test scores**

* A predefined list is used, e.g., [26, 49, 98, 87, 62, 75].

**4. Loop through each score in the list**

* For each score, the program checks a condition.

**5. Is the score even? (score % 2 == 0)**

* **Yes:** Add the score to sum\_of\_evens
* **No:** Skip to the next score

**Repeat Steps 4–5 until all scores are checked**

**6. Display the value of sum\_of\_evens**

* Shows the total sum of all even test scores.

**7. End**

* Program finishes.

A diagram of a diagram

AI-generated content may be incorrect. **Figure 1 Screenshot of flowcha****rt**

A screenshot of a computer

AI-generated content may be incorrect.   
 **Figure 2 Screenshot of sourcecode**  
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.

# Conclusion

This laboratory activity provided a practical understanding of how algorithms and flowcharts can be used to solve problems programmatically. In this task, we developed an algorithm that processes a list of test scores and calculates the sum of only the even numbers. The logic involved checking each score using a loop and conditional statement, and adding it to a running total if the score was even.  
  
To complement the algorithm, we created a flowchart that visually represented the step-by-step process—from reading each score to evaluating its parity and updating the total sum. This visual aid reinforced our understanding of how control structures like loops and conditionals operate within a program.  
  
The Python implementation worked effectively, accurately summing the even test scores and demonstrating how easily the code can be adapted for different datasets. Overall, this lab highlighted the importance of algorithm design, logical thinking, and planning through flowcharts before coding, which are all essential skills in programming.

**References**

1.Gaddis, T. (2022). *Starting out with Python* (5th ed.). Pearson Education.  
2. Dromey, R. G. (2006). *How to Solve it by Computer*. Pearson Education.  
3. Python Software Foundation. (2024). *Python 3 Documentation*. https://docs.python.org/3/