Data Structure and Algorithm

Laboratory Activity No. 3

Translating Algorithm to Program

|  |  |
| --- | --- |
| *Submitted by:* | *Instructor:* |
| Caasi, Karl Benedict D | Engr. Maria Rizette H. Sayo |

August 23, 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 Lucid Chart 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:

A diagram of a graph

AI-generated content may be incorrect.

Figure 1 Screenshot of program

This flowchart helps count how many odd numbers are in a list. It starts by setting a number called count\_ of\_ odds to zero. This number will keep track of how many odd scores we find.

Then, it checks each score in the list one by one. If a score is odd (not divisible by 2), it adds 1 to count\_ of\_ odds. If the score is even, it just skips it.

After checking all the scores, the flowchart shows the total number of odd scores by printing count \_of\_ odds. Finally, the program ends.

A screenshot of a computer program

AI-generated content may be incorrect.

Figure 2 Screenshot of program  
  
  
  
  
  
The code counts how many numbers in the list are odd by checking each number one by one, increasing a counter when a number is odd, and then prints the total count.  
  
  
  
  
  
  
  
  
Conclusion   
  
This task used a flowchart and Python code to count odd numbers in a list by checking each value and updating a counter, showcasing basic programming skills like loops and conditionals in a straightforward way.

REFERENCE  
Python documentation: https://docs.python.org/3/tutorial/controlflow.html#for-statements