Nanyang Technological University



**MA4830 – Realtime Software for Mechatronic Systems**

Minor Programming Assignment

Supervisor:

Prof. Gerald Seet

Student Name:

Jin Zihang U1822185F

Bryant U1820821E

Cai Yuxin U1822214D

Dylan Yeo U1922111H

Table of Contents

[Description 3](#_Toc84586974)

[Features 3](#_Toc84586975)

[Highlight 3](#_Toc84586976)

[Limitation 3](#_Toc84586977)

[Flow Chart 4](#_Toc84586978)

[Program Listing 5](#_Toc84586979)

[Process controls 5](#_Toc84586980)

[Example Run 8](#_Toc84586981)

[Directory Tree 10](#_Toc84586982)

# Description

We wrote a C program to calculate properties of standard 2D & 3D objects. The program will guide the user to choose and input the appropriate parameters as shown in **Table 1** and then output result will be tabulated and displayed onto the computer screen. This program can be executed multiple times with different shapes and the calculated results will be saved in the pre-allocated memory variable and will be presented in a table format with the mean and standard deviation values as requested.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Keyboard Input | | | 🡺 | Output Result | | | | Display  Calculation history, Mean & SD |
| Class of object | Type of object | Unit | Circumference | Area | Surface area | Volume |
| 2D Objects | Rectangle | m  dm  cm  mm | √ | √ |  |  |
| Square | √ | √ |  |  |
| Circle | √ | √ |  |  |
| 3D Objects | Cuboid |  |  | √ | √ |
| Cube |  |  | √ | √ |
| Sphere |  |  | √ | √ |
| Cone |  |  | √ | √ |

*Table 1. Input and Output chart*

## Features

1. Allow users to select “Unit” for calculation (m, dm, cm, mm) and automatically execute unit conversions and present requested calculation result in table format.
2. Allow users to do multiple calculations in sequence without exiting the program.
3. User can real-time check the input parameters they have.
4. User can backtrack when choosing shape to change the dimension of the shape that they would like.
5. Allow users to “Exit” programme when selecting shape or after calculation.
6. The programme is also able to identify input errors and give out case by case help.
7. Ability to calculate multiple shapes and calculate means and standard deviations.
8. Ability to display all previous calculation history in table format after every calculation.
9. Clean output display with usage of “=” and every round of calculations are separated neatly for ease of reading.

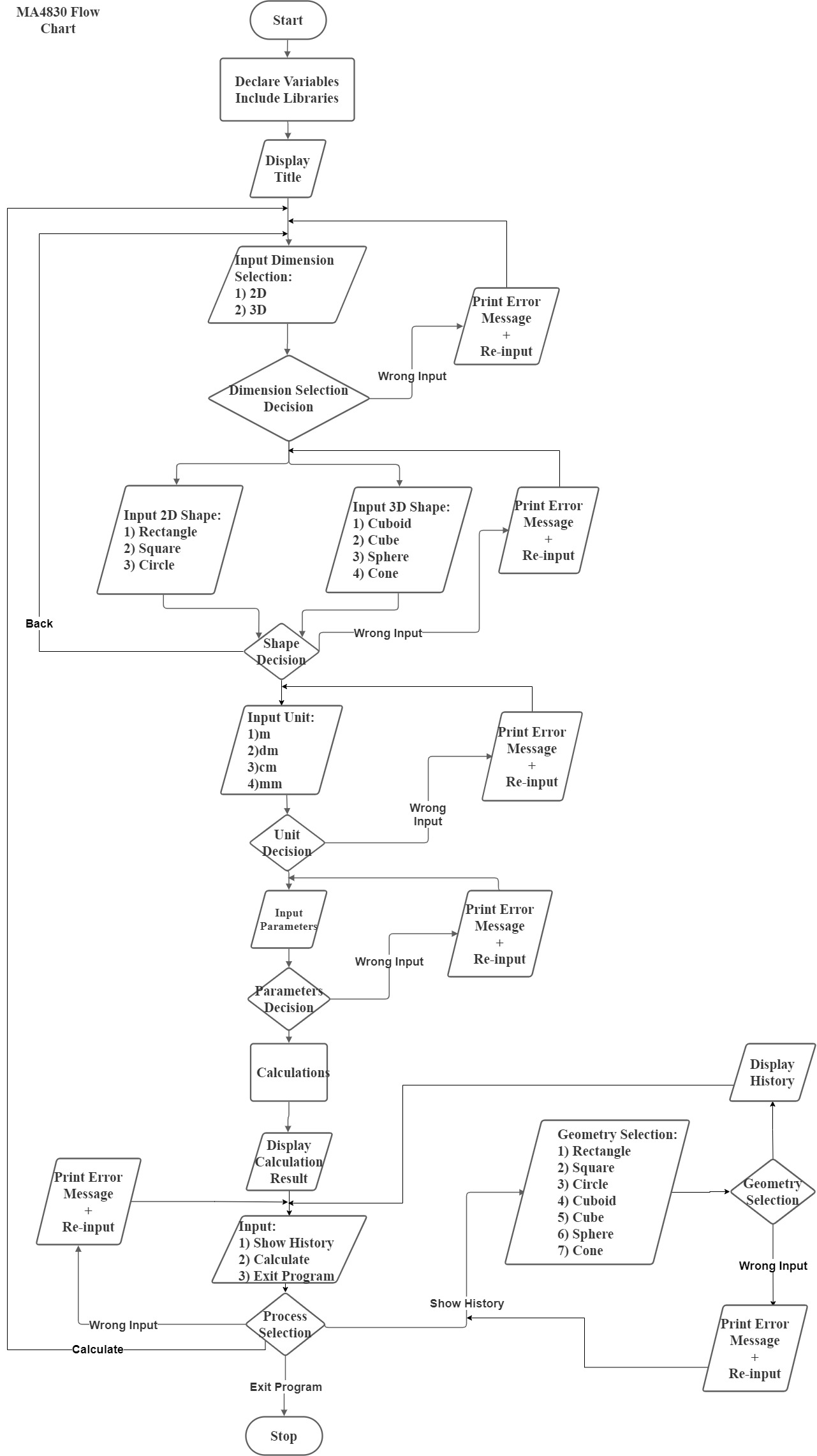
## Highlight

1. Multiple files which contain different functions for modularity purposes and easier-to-understand logic.
2. Enum and Struct to encapsulate a group of data with similar properties.
3. Usage of dynamic memory allocation to store the user’s input and will be freed after usage.

## Limitation

1. The variables (Length, Width, Area, Volume, etc.) are defined as doubles. If input max length and width to calculate area, it will exceed the memory allocation of the programme.
2. The history table only contains a maximum of 10 recently calculated data for each shape.

# **Flow** Chart



*Table 2. Overall Flowchart*

# Program Listing

The below Table shows all self-defined functions in the program and are separated into 4 categories. And below shows some key functions programming logic and method used to improve the ease-of-use and robustness of whole program.

|  |  |
| --- | --- |
| Input & Process controls | Text  Description automatically generated |
| Help & Suggestions printed |  |
| Calculation & Visualization |  |
| History and Mean & SD Display |  |

*Table 3. Categories of all functions*

## Process controls

1. The program logic of self-defined function **UnitSelection()** [**ProcessSelection()** is similar to this logic].

**Diagram

Description automatically generated**

*Table 4. Programming Logic for UnitSelection()*

1. The program logic of self-defined function: **GeometrySelection()**, **ShapeSelection()** and **ObjectSelection()**.

Diagram

Description automatically generated

*Table 5. Programming Logic for GeometrySelection()*

1. The program logic of self-defined function **DimensionSelection()**.

**Diagram

Description automatically generated**

*Table 6. Programming Logic for DimensionSelection()*

# Example Run

|  |  |
| --- | --- |
| Screenshots | Description |
| Graphical user interface, text  Description automatically generated | * Main title screen * 3 choices: 1)” 2D”/”1” 2)”3D”/”2” 3)”Exit” |
|  | * 2D shapes: 1) Rectangle 2) Square 3) Circle |
|  | * 3D shapes: 1) Cuboid 2) Cube 3) Sphere 4) Cone |
| A screenshot of a computer  Description automatically generated with medium confidence | * Error message will be sent if there is an invalid input |
|  | * “Enter positive number” message sent if input is “0” or negative number |
| Text  Description automatically generated | * “Back” command to change dimension of shape |
|  | * Input Units: 1) m 2) dm 3) cm 4) mm |

|  |  |
| --- | --- |
| A screenshot of a computer  Description automatically generated with medium confidence | * Request for parameters (Width, Length) * Display of calculation results in table form * Select: 1) History 2) Calculate 3) Exit   after calculation |
| A picture containing diagram  Description automatically generated | * Select history shapes: 1) Rectangle 2) Square 3) Circle 4) Cuboid 5) Cube 6) Sphere 7) Cone * Display of 2D shape history in table form * Mean and SD calculated |
| A screenshot of a computer  Description automatically generated with medium confidence | * Display of 3D shape history in table form * Mean and SD calculated |

# Directory Tree

Text

Description automatically generated

|  |  |
| --- | --- |
| **Files** | **Description** |
| Main.c | Execute Main programme |
| Title.txt | Contains Main Screen of geometry calculator |
| Enum.h | Contains enum of shapes and units |
| Struct.h | Contains parameters and their respective data types |
| Calculation.h | Contains formulas for calculation |
| Print.h | To print tables and the relevant results |
| Selection.h | To give the user choices for different inputs |

*Table 7. Descriptions of file directories*