

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
Features	3
Highlight.....	3
Limitation	3
Flow Chart.....	4
Program Listing	5
Process controls	5
Example Run	7
Directory Tree	9

Description

We wrote a C program to calculate the properties of standard 2D & 3D objects. The program guides users to choose and input appropriate parameters as shown in **Table 1**, then the output results will be tabulated and displayed onto the computer screen. This program can be executed multiple times for different geometries and the calculation results are stored in a history variable. The history table can be displayed as requested with means and standard deviations at the end of each column.

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. Users can real-time check the input parameters they have.
4. Users can backtrack while choosing a geometry to change the dimension of the geometry that they would like.
5. Allow users to “Exit” program when selecting a geometry or after calculation.
6. The program is also able to identify input errors and give out case-by-case help.
7. Ability to calculate properties for multiple geometries 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 the usage of “=” and every round of calculations are separated neatly for ease of reading.

Highlight

1. Multiple files 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 program.
2. The history table only contains a maximum of 10 recently calculated data for each geometry.

Flow Chart

MA4830 Flow
Chart

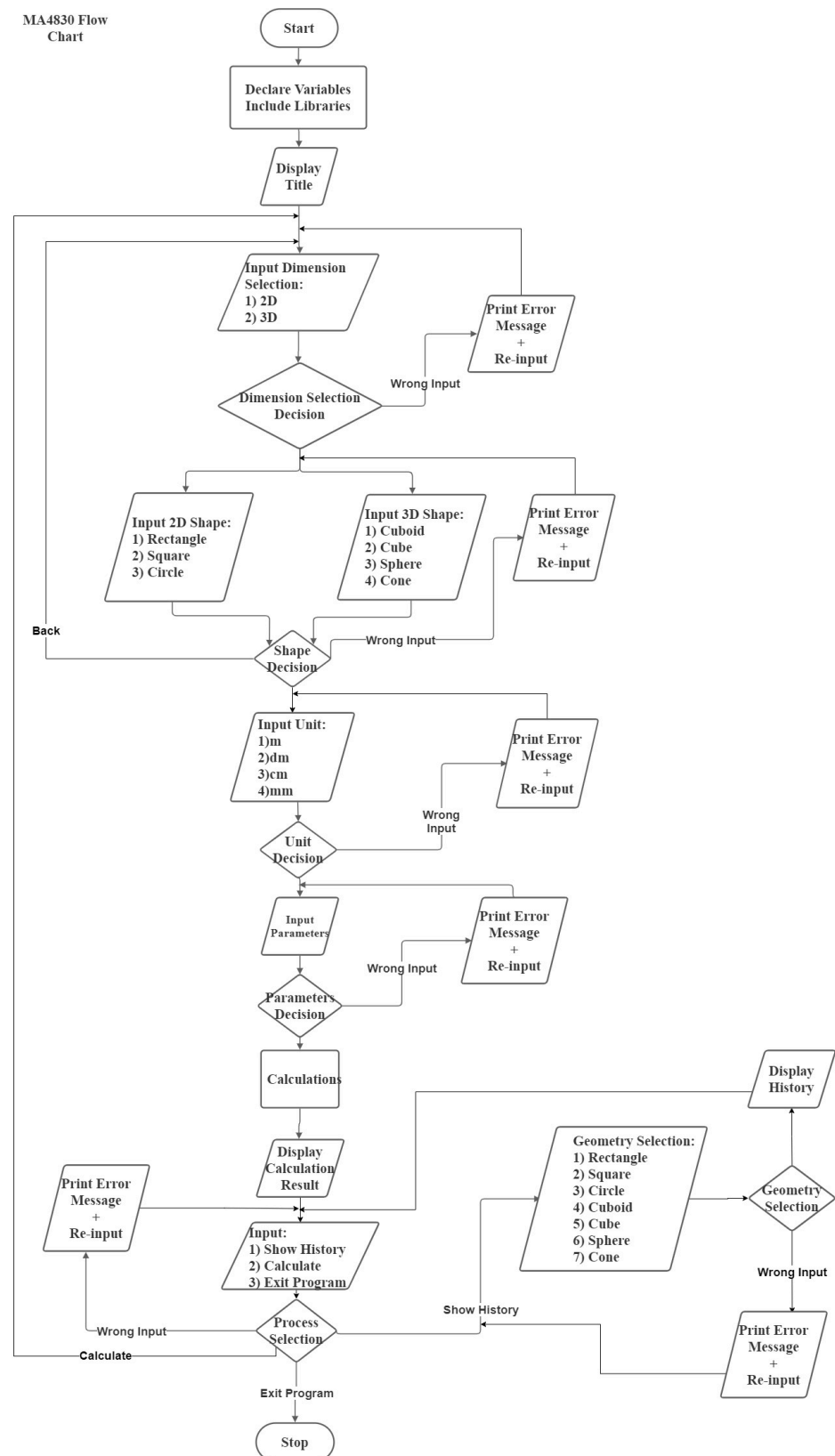


Table 2. Overall Flowchart

Program Listing

The **Table 3** below 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 the whole program.

Input & Process controls	<pre> > bool ShapeSelection(enum shape *shape) ... > bool ObjectSelection(enum shape *shape) ... > bool GeometrySelection(enum shape *shape, int dimension) ... > void DimensionSelection(int *dimension) ... > void UnitSelection(enum unit *unit) ... > bool ProcessSelection() ... > void ShapeAndObjectSelection(enum shape *shape) ... </pre>
Help & Suggestions printed	<pre> > void DisplayImage(FILE *fptr) ... > void DisplayTitle(char *filename) ... > void NoMemoryAlert() ... > void DimensionSelectionInstructions() ... > void WrongDimensionInput() ... > void ShapeSelectionInstructions() ... > void WrongShapeInput() ... > void ObjectSelectionInstructions() ... > void WrongObjectInput() ... > void UnitSelectionInstructions() ... > void WrongUnitInput() ... > void DisplayResults(enum shape shape, double result_1, double result_2) ... > void ParameterSelectionInstructions(char *parameter) ... > void NumericInputAlert(bool isnumeric) ... > void ProcessSelectionInstructions() ... > void WrongProcessInput() ... > void ShapeAndObjectSelectionInstructions() ... > void WrongShapeAndObjectInput() ... > void DisplayHistoryTable(enum shape shape, struct History *history, double *means, double *stds) ... </pre>
Calculation & Visualization	<pre> double GetParameterInput(void (*paramInstructions)(char *parameter), char *parameter) ... void AssignRectangleParameter(struct History *history, int base) ... void GetRectangleParameter(struct History *history, enum unit *unit) ... void AssignSquareParameter(struct History *history, int base) ... void GetSquareParameter(struct History *history, enum unit *unit) ... void AssignCircleParameter(struct History *history, int base) ... void GetCircleParameter(struct History *history, enum unit *unit) ... void AssignCuboidParameter(struct History *history, int base) ... void GetCuboidParameter(struct History *history, enum unit *unit) ... void AssignCubeParameter(struct History *history, int base) ... void GetCubeParameter(struct History *history, enum unit *unit) ... void AssignSphereParameter(struct History *history, int base) ... void GetSphereParameter(struct History *history, enum unit *unit) ... void AssignConeParameter(struct History *history, int base) ... void GetConeParameter(struct History *history, enum unit *unit) ... void CalculateProperties(enum shape shape, struct History *history) ... </pre>
History and Mean & SD Display	<pre> > void CalculateHistoricalProperties(struct History *history) ... </pre>

Table 3. Categories of all functions

Process controls

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

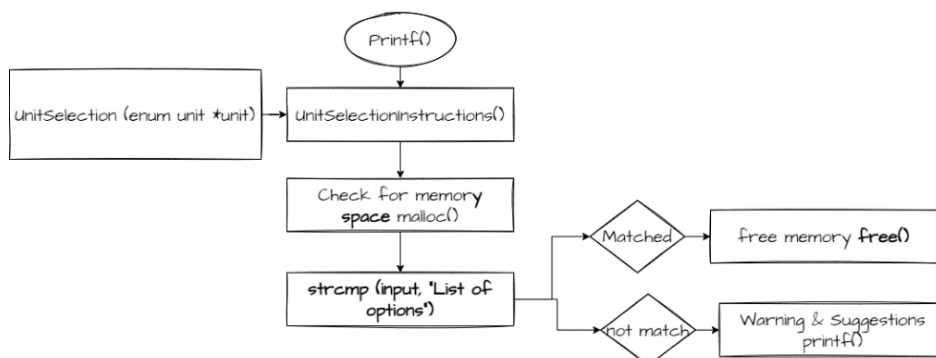


Table 4. Programming Logic for UnitSelection()

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

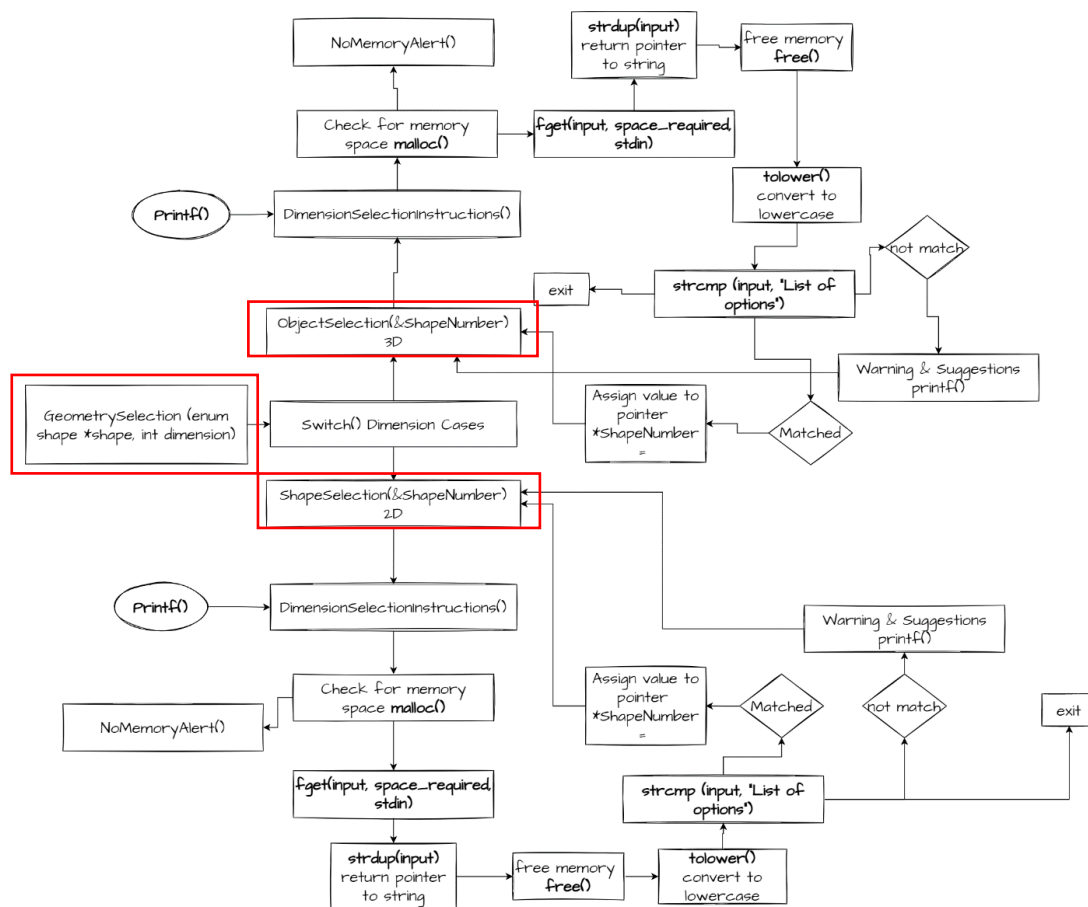


Table 5. Programming Logic for GeometrySelection()

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

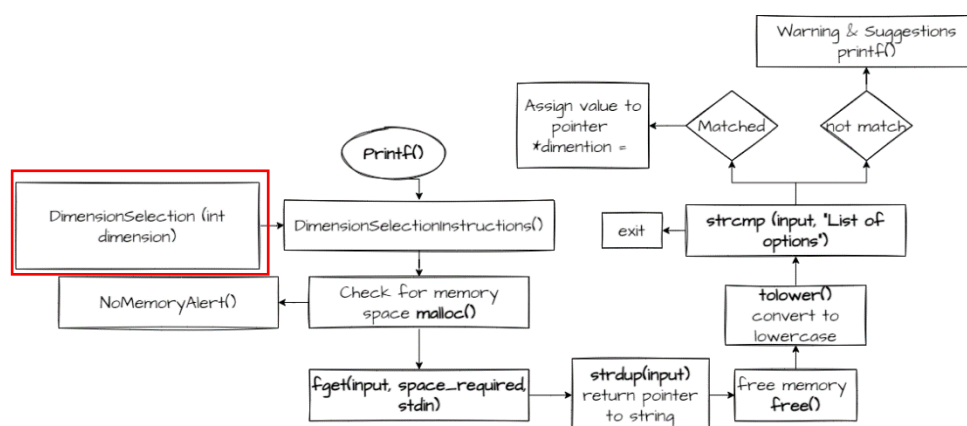
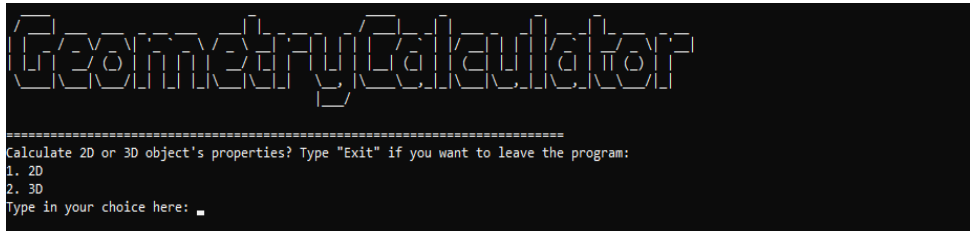
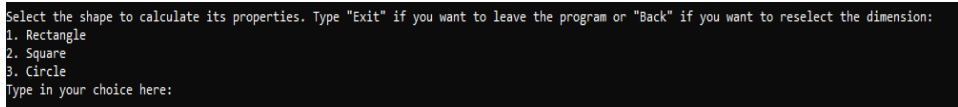
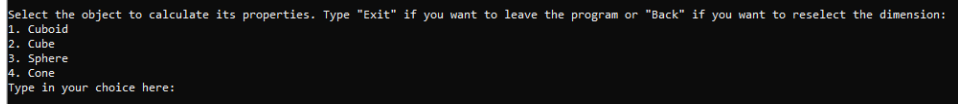
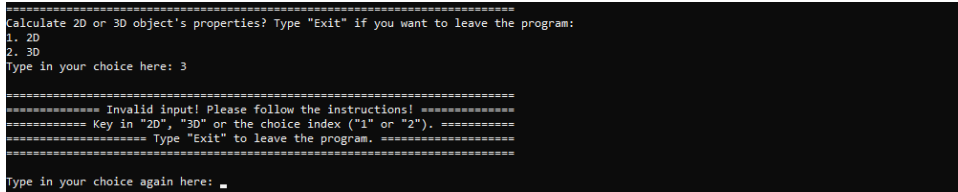
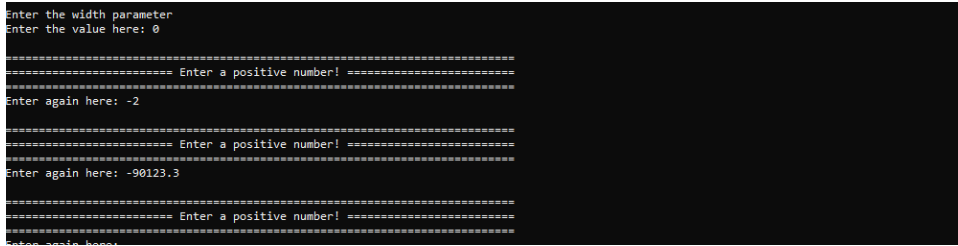
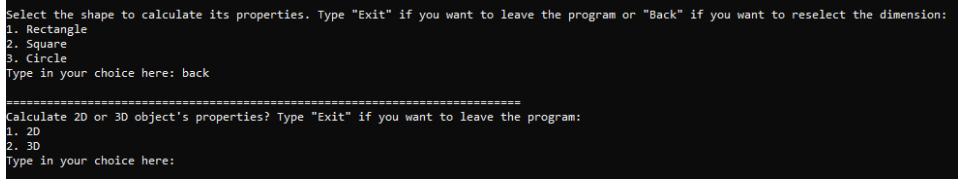



Table 6. Programming Logic for DimensionSelection()

Example Run

Screenshots	Description
	<ul style="list-style-type: none"> Main title screen 3 choices: 1)" 2D"/" 1" 2)" 3D"/" 2" 3)"Exit"
	<ul style="list-style-type: none"> 2D shapes: 1) Rectangle 2) Square 3) Circle
	<ul style="list-style-type: none"> 3D objects: 1) Cuboid 2) Cube 3) Sphere 4) Cone
	<ul style="list-style-type: none"> An error message will be sent if there is an invalid input
	<ul style="list-style-type: none"> "Enter positive number" message sent if the input is "0" or negative number
	<ul style="list-style-type: none"> "Back" command to change the dimension of geometry
	<ul style="list-style-type: none"> Input Units: 1) m 2) dm 3) cm 4) mm

Enter the width parameter
Enter the value here: 2

Enter the length parameter
Enter the value here: 3

Calculation results:

Perimeter	10 m	1e+002 dm	1e+003 cm	1e+004 mm
Area	6 m^2	6e+002 dm^2	6e+004 cm^2	6e+006 mm^2

Select:

1. History - To view the calculation history.
2. Calculate - To calculate again.
3. Exit - To leave the program.

Enter your choice here: █

- Request for parameters (Width, Length)
- Display of calculation results in table form
- Select:
 - 1) History
 - 2) Calculate
 - 3) Exit
 after calculation

Select any of the option

1. Rectangle
2. Square
3. Circle
4. Cuboid
5. Cube
6. Sphere
7. Cone

Type in your choice here: 2

Calculation Histroy of Square

Index	Side Lenth	Perimeter	Area
0	4 m	16 m	16 m^2
1	10 m	40 m	1e+002 m^2
Mean	7 m	28 m	58 m^2
Standard Deviation	3 m	12 m	42 m^2

Select:

1. History - To view the calculation history.
2. Calculate - To calculate again.
3. Exit - To leave the program.

Enter your choice here: █

- Select history geometries:
 - 1) Rectangle
 - 2) Square
 - 3) Circle
 - 4) Cuboid
 - 5) Cube
 - 6) Sphere
 - 7) Cone
- Display of 2D shapes' calculation history in table form
- Mean and SD calculated

Select any of the option

1. Rectangle
2. Square
3. Circle
4. Cuboid
5. Cube
6. Sphere
7. Cone

Type in your choice here: 6

Calculation Histroy of Sphere

Index	Radius	Surface Area	Volume
0	6 m	4.5e+002 m^2	6.8e+002 m^3
1	10 m	1.3e+003 m^2	3.1e+003 m^3
2	14 m	2.5e+003 m^2	8.6e+003 m^3
Mean	10 m	1.4e+003 m^2	4.1e+003 m^3
Standard Deviation	3.3 m	8.3e+002 m^2	3.3e+003 m^3

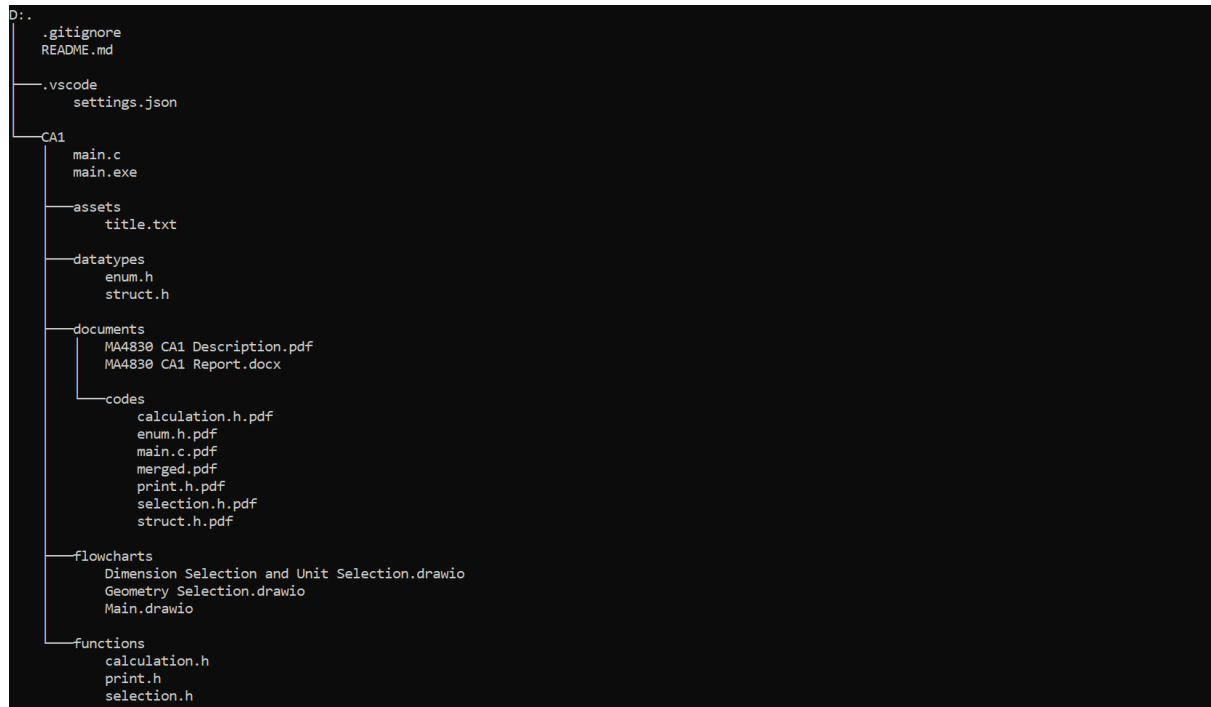
Select:

1. History - To view the calculation history.
2. Calculate - To calculate again.
3. Exit - To leave the program.

Enter your choice here:

- Display of 3D objects' calculation history in table form
- Mean and SD calculated

Directory Tree



Files	Description
Main.c	Execute Main programme
Title.txt	Contains Main Screen of geometry calculator
Enum.h	Contains enum of geometries 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