# Assignment Description

In this program, you will create a cube object and display its volume. Create a class Cube that includes private data members length (int), width (int), height (int), and color (string).

Create a constructor for the class that receives the length, width, height, and color. The constructor should set the attributes with the values provided. Your class should also have public member functions:

* Accessor methods getLength, getWidth, getHeight, and getColor returning the appropriate attribute of the cube.
* Mutator methods setLength, setWidth, setHeight, and setColor that allow the attributes of the cube to be changed.
* A calculateVolume method that will return the volume of the cube (volume = length\*width \* height).

Create a main program that utilizes the Cube class

1. Prompt the user for the length, width, height, and color of a cube
2. Validate the information
3. After Information from the user has been validated, create an object for the Cube.
4. Using the Accessor Methods, display the cube’s information (length, width, height, color, and volume).
5. Allow the user to change/set the length, width, height, and color. Validate any input before calling the appropriate functions.
6. Display the cube’s attributes after the cube has been changed using the appropriate methods.

# 1 Readme Documentation

This program lets the user build a cube, inputting its dimensions and color. Then, the program will output the characteristics of the cube.

# 2 Flowchart Screen Shots

# 3 UML and Use Case Diagrams

# 4 Source Code of All files (.h, .cpp)

#include *<iostream>*

#include *<iomanip>*

#include *<string>*

#include *<cctype>*

**using** **namespace** **std**;

*/\**

*Name: Cube Constructor*

*Author: Wesley Hixon*

*Date Last Updated: 10/22/2024*

*Purpose: Build a Cube class with the attributes length, width, height, and color,*

*as well as methods to set and get the attributes. Then, allow the user to build a*

*cube through setting the attributes of a new Cube object.*

*\*/*

*// Cube class, includes length, width, height, and color*

**class** **Cube**{

**private**:

int length;

int width;

int height;

string color;

**public**:

*// Methods to return attributes*

int getLength(){

**return** length;

}

int getWidth(){

**return** width;

}

int getHeight(){

**return** height;

}

string getColor(){

**return** color;

}

*// Methods to set attributes*

void setLength(int inputLength){

length = inputLength;

}

void setWidth(int inputWidth){

width = inputWidth;

}

void setHeight(int inputHeight){

height = inputHeight;

}

void setColor(string inputColor){

color = inputColor;

}

*// Method to calculate volume (l\*w\*h)*

double calculateVolume(){

double volume = length\*width\*height;

**return** volume;

}

};

*// Function to build a new cube*

Cube constructCube();

*// Function to output cube characteristics*

void readCubeInfo(Cube cube);

void validateInt(int& input);

int main(){

*// Create new cube*

Cube userCube;

*// Build new cube with constructCube()*

userCube = constructCube();

*// Read back cube info*

readCubeInfo(userCube);

**return** 0;

}

*// This function asks for input for cube characteristics and returns the final cube*

Cube constructCube(){

*// Initializing variables*

int length, width, height;

string color;

Cube newCube;

*// Length input*

cout << "Please enter the length of your cube: ";

validateInt(length);

*// Setting length*

newCube.setLength(length);

*// Width input*

cout << "Please enter the width of your cube: ";

validateInt(width);

*// Setting width*

newCube.setWidth(width);

*// Height input*

cout << "Please enter the height of your cube: ";

validateInt(height);

*// Setting height*

newCube.setHeight(height);

*// Color input*

cout << "Please enter the color of your cube: ";

*// Validating color input*

bool valid = false;

bool containsNumber = false;

**while**(!valid){

cin >> color;

*// In case of input failure*

**if**(!cin){

cout << "Please try again.";

cin.clear();

cin.ignore(10000, '\n');

}

**else**{

*// Checks every character for numbers*

containsNumber = false;

**for**(int i = 0; i < color.length(); i++){

**if**(!isalpha(color[i])){

cout << "Please enter a string without numbers.";

containsNumber = true;

**break**;

}

}

}

**if**(!containsNumber){

valid = true;

}

}

*// Setting color*

newCube.setColor(color);

*// Returning newCube*

**return** newCube;

}

*// Function to output cube info to console using getter methods*

void readCubeInfo(Cube cube){

cout << endl << "This is your cube info" << endl

<< "The length of your cube is " << cube.getLength() << endl

<< "The width of your cube is " << cube.getWidth() << endl

<< "The height of your cube is " << cube.getHeight() << endl

<< "The volume of your cube is " << cube.calculateVolume() << endl

<< "The color of your cube is " << cube.getColor() << endl;

**return**;

}

*// Function to prompt for a valid integer input over 0*

void validateInt(int& input){

**while**(!(cin >> input) || input <= 0){

cout << "Try again. Please enter a valid integer.";

cin.clear();

cin.ignore(10000, '\n');

}

}

# 5 Three Use Case Screen Shots

