# Assignment Description

You are working on a program that helps bakers convert from weight measurements to volume measurements and back again.

# GitHub URL (optional)

[https://github.com/wesleyhixon/Programming-Assignments/tree/cd2ffbe40d425074dc111adc3443020cba6520df/M05%20Programming%20Assignment%202](https://github.com/wesleyhixon/Programming-Assignments/tree/cd2ffbe40d425074dc111adc3443020cba6520df/M05 Programming Assignment 2)

# Readme Documentation

Input Information:Ingredient name, whether to convert from weight or volume, and then a weight or volume

Output Information: A given weight converted to volume or vice versa

# Flowchart Screen Shots (optional)

Screen shot(s) here

# UML and Use Case Diagrams (optional)

Screen shot(s) here

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

#include *<iostream>*

#include *<map>*

#include *<iomanip>*

#include *<list>*

#include *<bits/stdc++.h>*

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

*/\**

*Program Name: Volume and Mass*

*Author: Wesley Hixon*

*Date Last Updated: 07/03/2024*

*Purpose: Convert ingredient weight or volume to volume or weight.*

*\*/*

*// Declaring ingredientDensities map so that it's global*

map<string, double> ingredientDensities;

*// This is a function that converts a string to lowercase*

*// std::tolower only works on chars so I have to use transform to iterate the entire string*

string to\_lowercase(string input){

transform(input.begin(), input.end(), input.begin(), ::tolower);

**return** input;

}

string getIngredient(){

string ingredient;

bool valid = false;

*// Prompt the user for the ingredient they want to convert*

**while**(!valid){

*// Get input*

cout << "Enter the name of the ingredient: ";

getline(cin, ingredient);

*// Transform input to lowercase*

ingredient = to\_lowercase(ingredient);

*// If the ingredient is found in ingredientDensities map, input is valid*

**if**(ingredientDensities.find(ingredient) != ingredientDensities.end()){

valid = true;

}

*// In case of input failure*

**else** **if**(!cin){

cout << "I don't know that ingredient. Here is a list of ingredients I do know:" << endl;

cout << "Sugar, Flour, Cocoa Powder, Honey, Egg, Milk, Water, and Oil" << endl;

cin.clear();

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

}

*// Otherwise, input is invalid*

**else**{

cout << "I don't know that ingredient. Here is a list of ingredients I do know:" << endl;

cout << "Sugar, Flour, Cocoa Powder, Honey, Egg, Milk, Water, and Oil" << endl;

}

}

**return** ingredient;

}

char weightOrVolume(){

*// This function prompts for weight or volume and validates the input*

bool valid = false;

char wOrV;

**while**(!valid){

cout << "Do you want to convert from weight or volume? (Enter W or V): ";

cin >> wOrV;

**if**(wOrV == 'w' || wOrV == 'W'){

**return** 'w';

}

**else** **if**(wOrV == 'v' || wOrV == 'V'){

**return** 'v';

}

**else** **if**(!cin){

cout << "Try again.";

cin.clear();

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

}

**else**{

cout << "Try again.";

}

}

**return** '0';

}

double getDensity(string ingredientName){

*// Given the ingredient name, get the density from the densities map*

*// This one isn't really necessary but I included it because the assignment said it's required*

**return** ingredientDensities[ingredientName];

}

double getWeight(){

*// prompt the user for a weight and validates the input*

bool valid = false;

double weight;

**while**(!valid){

cout << "Enter the weight of your ingredient in grams: ";

cin >> weight;

**if**(weight < 0){

cout << "Please enter a number over 0." << endl;

}

**else** **if** (!cin){

cout << "Please enter a valid number." << endl;

cin.clear();

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

}

**else**{

valid = true;

}

}

**return** weight;

}

double getVolume(){

*// prompt the user for a volume and validates the input*

bool valid = false;

double volume;

**while**(!valid){

cout << "Enter the volume of your ingredient in milliliters: ";

cin >> volume;

**if**(volume < 0){

cout << "Please enter a number over 0." << endl;

}

**else** **if** (!cin){

cout << "Please enter a valid number." << endl;

cin.clear();

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

}

**else**{

valid = true;

}

}

**return** volume;

}

double convertWeight(double weight, double density){

*// Given weight and density, calculate the volume*

double volume;

volume = weight / density;

**return** volume;

}

double convertVolume(double volume, double density){

*// Given the volume and density, calculate the weight*

double weight;

weight = volume \* density;

**return** weight;

}

int main()

{

*// Defining variables*

string userIngredient;

char userChoice;

double weight, volume, density;

*// Welcoming the user to my humble program*

cout << "Welcome to the baking converter." << endl;

cout << "This program takes a baking ingredient and converts a weight to a volume unit or a volume to a weight unit." << endl;

*// Defining the ingredientDensities map for future use*

ingredientDensities = {

{"sugar", 1.59},

{"flour", 0.762},

{"cocoa powder", 1.35},

{"honey", 1.38},

{"egg", 1.031},

{"milk", 1.025},

{"water", 0.997},

{"oil", 0.91}

};

*// Getting the ingredient the user wishes to convert*

userIngredient = getIngredient();

*// Getting whether the user wants to convert from weight or volume*

userChoice = weightOrVolume();

**if**(userChoice == 'w'){

*// If user chooses weight, get the weight*

weight = getWeight();

*// Get ingredient density*

density = getDensity(userIngredient);

*// Convert to volume*

volume = convertWeight(weight, density);

*// Output the final conversion*

cout << fixed << showpoint << setprecision(4) << weight << " g of " << userIngredient << " is " << volume << " ml." << endl;

}

**else** **if** (userChoice == 'v'){

*// If the user chooses volume, get the volume*

volume = getVolume();

*// Get ingredient density*

density = getDensity(userIngredient);

*// Convert to weight*

weight = convertVolume(volume, density);

*// Output the final conversion*

cout << fixed << showpoint << setprecision(4) << volume << " ml of " << userIngredient << " is " << weight << " g." << endl;

}

**return** 0;

}

# Three Use Case Screen Shots





