Coffe Machine Project

Enita Mujkanović, CS103, 2022.

Hello professor, before you get into this program, I'd like to make a quick note.

I've had help from a friend who passed Intro. to Programming last semester, he is very skilled with C++ and C and he was of great help. I learned a lot of stuff while doing this project that I thought I will never quite catch how it is done.

Now on to the project, I know you said we needed to do it in different classes, but we did it in one since it was a little confusing for both of us, and we couldn't find good enough explanations on the internet regarding the use of multiple files and classes (that is why I asked of You to post the files regarding classes).

Also, regarding the choosing of the type of drink and the amount of funds, please beware of the amount of money that is already inside a coffee machine, and try to enter the amount that the machine will be able to give You back. I think that is all for this little note.

Enjoy!

## **CoffeeMachine.h**

#ifndef COFFEEMACHINE\_H

#define COFFEEMACHINE\_H

#include <string>

using namespace std;

class CoffeeMachine

{

public:

CoffeeMachine();

double moneyEntered(double digit);

//

void welcomeScreen();

void showQuantities();

void selection(char query, double& money);

void checkMoney();

void selectionScreen();

bool returnCorrectValue(double money);

void recompile(double money);

void decrementQuantity(char query);

void insufficient(double money);

bool askToContinue(char query);

void checkQuantity();

//

void showMachineMoney();

//

int getEspressoQ();

int getMacchiatoQ();

int getHotChocolateQ();

int getCappuccinoQ();

private:

double espressoPrice = 0.5;

double macchiatoPrice = 1.0;

double hotChocolatePrice = 1.5;

double cappuccinoPrice = 2.0;

int espressoQ, macchiatoQ, hotChocolateQ, cappuccinoQ;

int numberOfHalfs = 0;

int numberOfMarks = 7;

int numberOfTwoMarks = 0;

int numberOfFiveMarks = 10;

};

#endif // COFFEEMACHINE\_H

## **Main**

#include <iostream>

#include <string>

#include "CoffeeMachine.h"

using namespace std;

int main()

{

CoffeeMachine machine1;

char query;

machine1.welcomeScreen();

do {

double money;

char selection;

machine1.checkMoney();

cin >> money;

machine1.recompile(money);

machine1.selectionScreen();

cin >> selection;

machine1.checkQuantity();

machine1.selection(selection, money);

machine1.showQuantities();

machine1.showMachineMoney();

cout << "Displaying your current balance: " << money << " KM." << endl;

cout << endl;

cout << "Do you want to exit the program? (Y/N): ";

cin >> query;

if(machine1.askToContinue(query))

continue;

else {

cout << "\n--> PROGRAM ENDED <---";

break;

}

} while(true);

return 0;

}

## **Coffe Machine**

#include "CoffeeMachine.h"

#include <iostream>

using namespace std;

// default constructor

CoffeeMachine::CoffeeMachine()

/\*Here is where the constructor starts being used. We named it CoffeMachine.h, he sets the values of everything at 50, so everytime the program starts again, it will return the quantity of items at 50 again.\*/

{

espressoQ = 50;

macchiatoQ = 50;

hotChocolateQ = 50;

cappuccinoQ = 50;

}

int CoffeeMachine::getEspressoQ()

{

return espressoQ;

}

int CoffeeMachine::getMacchiatoQ()

{

return macchiatoQ;

}

int CoffeeMachine::getHotChocolateQ()

{

return hotChocolateQ;

}

int CoffeeMachine::getCappuccinoQ()

{

return cappuccinoQ;

}

// Function to show how many drinks there are left in the machine.

// returns: nothing

void CoffeeMachine::showQuantities()

{

cout << "\nDisplaying current quantity of drinks in the machine: " << endl;

cout << "-> Espresso quantity: " << getEspressoQ() << endl;

cout << "-> Macchiato quantity: " << getMacchiatoQ() << endl;

cout << "-> Hot chocolate quantity: " << getHotChocolateQ() << endl;

cout << "-> Cappuccino quantity: " << getCappuccinoQ() << endl;

}

// Function to display the selection screen.

// returns: nothing

void CoffeeMachine::selectionScreen()

{

cout << "\nSelect your drink: " << endl;

cout << "1. Espresso - 0.5 KM" << endl;

cout << "2. Macchiato - 1.0 KM" << endl;

cout << "3. Hot Chocolate - 1.5 KM" << endl;

cout << "4. Cappuccino - 2.0 KM" << endl;

cout << "\nPress a number -> ";

}

// Function to display the welcome screen

// returns: nothing

void CoffeeMachine::welcomeScreen()

{

// design decision?

cout << "...::: WELCOME TO THE COFFEE MACHINE BY E.M. :::...\n" << endl;

cout << "1. TO USE THE MACHINE, INSERT MONEY AND SELECT YOUR TYPE OF COFFEE" << endl;

cout << "2. BE AWARE THAT MACHINE CAN BE OUT OF SERVICE" << endl;

cout << "3. BE AWARE OF A MACHINE NOT BEING ABLE TO DISPENSE THE CHANGE" << endl;

cout << "4. IF ANY ERROR HAPPENS, MACHINE DISPENSES YOUR MONEY\n" << endl;

}

// Main logic of the program

// Function manipulates all member variables

// returns: nothing

/\*For this part of program, we have tried every single combination we could think of to break the program and see where we could improve it so the calculations could end up being correct. We could have done it the easier way by using a different method, but we decided last minute on **if, else if** since it was the only one we were 100% sure we knew how to use without screwing it up.

\*If you do manage to break the program, please do tell me so we can add more of code and fix it. You are welcome to try!\*/

void CoffeeMachine::selection(char query, double& money)

{

if(query == '1') {

if(money == 1.0 && numberOfHalfs > 0) {

numberOfMarks++;

numberOfHalfs--;

decrementQuantity(query);

money -= 0.5;

}

else if((money == 2.0) && ((numberOfMarks >= 0) && (numberOfHalfs >= 0))) {

if((numberOfHalfs > numberOfMarks) && (numberOfHalfs > 2)) {

numberOfTwoMarks++;

numberOfHalfs -= 3;

decrementQuantity(query);

money -= 0.5;

}

else if(numberOfHalfs >= 3) {

numberOfTwoMarks++;

numberOfHalfs -= 3;

decrementQuantity(query);

money -= 0.5;

}

else {

numberOfTwoMarks++;

numberOfMarks--;

numberOfHalfs--;

decrementQuantity(query);

money -= 0.5;

}

}

else if((money == 5.0) && (numberOfHalfs > 0)) {

if((numberOfTwoMarks > numberOfMarks) && (numberOfTwoMarks > numberOfHalfs) && (numberOfTwoMarks >= 2)){

numberOfFiveMarks++;

numberOfTwoMarks -= 2;

numberOfHalfs--;

decrementQuantity(query);

money -= 0.5;

}

else if((numberOfMarks > numberOfTwoMarks) && (numberOfMarks > numberOfHalfs) && (numberOfMarks >=4) && (numberOfHalfs >= 1)) {

numberOfFiveMarks++;

numberOfMarks -= 4;

numberOfHalfs--;

decrementQuantity(query);

money -= 0.5;

}

else if((numberOfHalfs > numberOfTwoMarks) && (numberOfHalfs > numberOfMarks) && (numberOfHalfs >= 9)) {

numberOfFiveMarks++;

numberOfHalfs -= 9;

decrementQuantity(query);

money -= 0.5;

}

else if((numberOfMarks >= 4) && (numberOfHalfs >= 1)) {

numberOfFiveMarks++;

numberOfMarks -= 4;

numberOfHalfs--;

decrementQuantity(query);

money -= 0.5;

}

else if((numberOfTwoMarks >= 2) && (numberOfHalfs >= 1)) {

numberOfFiveMarks++;

numberOfTwoMarks -= 2;

numberOfHalfs--;

decrementQuantity(query);

money -= 0.5;

}

else if((numberOfTwoMarks >= 1) && (numberOfMarks >= 2) && (numberOfHalfs >= 1)){

numberOfFiveMarks++;

numberOfTwoMarks--;

numberOfMarks -= 2;

numberOfHalfs--;

decrementQuantity(query);

money -= 0.5;

}

else {

insufficient(money);

}

}

else if(money == 0.5) {

numberOfHalfs++;

decrementQuantity(query);

money -= 0.5;

}

else {

insufficient(money);

}

}

else if(query == '2') {

if(money == 1.0) {

numberOfMarks++;

decrementQuantity(query);

money -= 1.0;

}

else if(money == 2.0) {

if((numberOfMarks > 0) && (numberOfMarks > numberOfHalfs)) {

numberOfTwoMarks++;

numberOfMarks--;

decrementQuantity(query);

money -= 1.0;

}

else if((numberOfHalfs > 1) && (numberOfHalfs > numberOfMarks)) {

numberOfTwoMarks++;

numberOfHalfs -= 2;

decrementQuantity(query);

money -= 1.0;

}

else if((numberOfHalfs > 1) && (numberOfMarks == 0)){

numberOfTwoMarks++;

numberOfHalfs -= 2;

decrementQuantity(query);

money -= 1.0;

}

else if((numberOfMarks == numberOfHalfs) && (numberOfMarks > 0) && (numberOfHalfs > 0)){

numberOfTwoMarks++;

numberOfMarks--;

decrementQuantity(query);

money -= 1;

}

else {

insufficient(money);

}

}

else if(money == 5.0) {

if((numberOfHalfs >= numberOfMarks) && (numberOfHalfs > numberOfTwoMarks) && (numberOfHalfs >= 6) && (numberOfMarks >= 1)) {

numberOfHalfs -= 6;

numberOfMarks--;

decrementQuantity(query);

money -= 1;

}

else if(numberOfTwoMarks >= 2) {

numberOfFiveMarks++;

numberOfTwoMarks -= 2;

decrementQuantity(query);

money -= 1;

}

else if((numberOfMarks >= 4) && (numberOfMarks > numberOfTwoMarks) && (numberOfMarks > numberOfHalfs)) {

numberOfFiveMarks++;

numberOfMarks -= 4;

decrementQuantity(query);

money -= 1;

}

else if((numberOfHalfs >= 8) && (numberOfHalfs > numberOfMarks) && (numberOfHalfs > numberOfTwoMarks)) {

numberOfFiveMarks++;

numberOfHalfs -= 8;

decrementQuantity(query);

money -= 1;

}

else if((numberOfHalfs >= 2) && (numberOfMarks >= 1) && (numberOfTwoMarks >= 1)) {

numberOfHalfs -= 2;

numberOfTwoMarks--;

numberOfMarks--;

decrementQuantity(query);

money -= 1;

}

else if((numberOfTwoMarks >= 1) && (numberOfHalfs >= 4)) {

numberOfFiveMarks++;

numberOfTwoMarks--;

numberOfHalfs -= 4;

decrementQuantity(query);

money -= 2;

}

else {

insufficient(money);

}

}

else if(money == 0.5) {

insufficient(money);

}

else {

insufficient(money);

}

}

else if(query == '3') {

if(money == 2.0) {

if(numberOfHalfs > 0) {

numberOfTwoMarks++;

numberOfHalfs--;

decrementQuantity(query);

money -= 1.5;

}

else {

insufficient(money);

}

}

else if((money == 5) && (numberOfHalfs > 0)) {

if((numberOfHalfs >= 7) && (numberOfHalfs > numberOfMarks) && (numberOfHalfs > numberOfTwoMarks)) {

numberOfFiveMarks++;

numberOfHalfs -= 7;

decrementQuantity(query);

money -= 1.5;

}

else if((numberOfTwoMarks >= 1) && (numberOfHalfs > 0) && (numberOfMarks >= 1)){

numberOfFiveMarks++;

numberOfTwoMarks -= 1;

numberOfMarks--;

numberOfHalfs--;

decrementQuantity(query);

money -= 1.5;

}

else if((numberOfMarks >= 3) && (numberOfHalfs > 1)) {

numberOfFiveMarks++;

numberOfMarks -= 3;

numberOfHalfs--;

decrementQuantity(query);

money -= 1.5;

}

else if((numberOfTwoMarks >= 1) && (numberOfMarks >= 1) && (numberOfHalfs >= 1)) {

numberOfFiveMarks++;

numberOfTwoMarks--;

numberOfMarks--;

numberOfHalfs--;

decrementQuantity(query);

money -= 1.5;

}

else if((numberOfTwoMarks == 0) && (numberOfMarks >= 2) && (numberOfHalfs >= 3)) {

numberOfFiveMarks++;

numberOfMarks -= 2;

numberOfHalfs -= 3;

decrementQuantity(query);

money -= 1.5;

}

else if((numberOfTwoMarks == 0) && (numberOfMarks >= 1) && (numberOfHalfs >= 5)) {

numberOfFiveMarks++;

numberOfMarks--;

numberOfHalfs -= 5;

decrementQuantity(query);

money -= 1.5;

}

else if((numberOfMarks == 0) && (numberOfTwoMarks >= 1) && (numberOfHalfs >= 3)) {

numberOfTwoMarks--;

numberOfHalfs -= 3;

decrementQuantity(query);

money -= 1.5;

}

else {

insufficient(money);

}

}

else if(money == 1.0) {

insufficient(money);

}

else if(money == 0.5) {

insufficient(money);

}

else {

insufficient(money);

}

}

else if(query == '4') {

if(money == 2) {

if(numberOfTwoMarks >= 1){

numberOfTwoMarks++;

decrementQuantity(query);

money -= 2.0;

}

else {

insufficient(money);

}

}

else if(money == 5) {

if((numberOfTwoMarks >= 1) && (numberOfMarks >= 1)) {

numberOfFiveMarks++;

numberOfTwoMarks--;

numberOfMarks--;

decrementQuantity(query);

money -= 2.0;

}

else if((numberOfTwoMarks >= 1) && (numberOfHalfs >= 2)) {

numberOfFiveMarks++;

numberOfTwoMarks--;

numberOfHalfs -= 2;

decrementQuantity(query);

money -= 2.0;

}

else if(numberOfHalfs >= 6) {

numberOfFiveMarks++;

numberOfHalfs -= 6;

decrementQuantity(query);

money -= 2.0;

}

else if((numberOfHalfs >= 2) && (numberOfMarks >= 1) && (numberOfTwoMarks >= 1)) {

numberOfFiveMarks++;

numberOfHalfs -= 2;

numberOfMarks--;

numberOfTwoMarks--;

decrementQuantity(query);

money -= 2.0;

}

else if((numberOfTwoMarks == 0) && (numberOfMarks >= 2) && (numberOfHalfs >= 2)) {

numberOfFiveMarks++;

numberOfMarks -= 2;

numberOfHalfs -=2;

decrementQuantity(query);

money -= 2.0;

}

else if((numberOfTwoMarks == 0) && (numberOfMarks >= 1) && (numberOfHalfs >= 4)) {

numberOfFiveMarks++;

numberOfHalfs -= 4;

numberOfMarks--;

decrementQuantity(query);

money -= 2.0;

}

}

else if(money == 1.0){

insufficient(money);

}

else if(money == 0.5) {

insufficient(money);

}

}

else {

cout << "Fatal error occurred. Exiting program." << endl;

exit(-1);

}

}

// Function to check if money input is valid. It calls the returnCorrectValue(double money) bool function.

// If the bool value is true nothing happens. If the bool value is false, program ends.

// returns: nothing

void CoffeeMachine::recompile(double money)

{

if(!returnCorrectValue(money)) {

cout << "Invalid value. Re-compile program to try again. " << endl;

exit(-1);

}

}

// Function to NOTE and display the money insert message.

// returns: nothing

void CoffeeMachine::checkMoney()

{

cout << "\nNOTE: Inserted values can only be: 0.5 KM, 1 KM, 2 KM or 5 KM" << endl;

cout << "Insert money -> ";

}

// Function to check if entered money is valid or not

// returns: true if valid, false if not

bool CoffeeMachine::returnCorrectValue(double money)

{

bool query = false;

if(money == 0.5)

query = true;

else if(money == 1.0)

query = true;

else if(money == 2.0)

query = true;

else if(money == 5.0)

query = true;

else {

query = false;

}

return query;

}

// Function to decrement the value by 1 with respect to the passed argument

// returns: nothing

void CoffeeMachine::decrementQuantity(char query)

{

if(query == '1')

espressoQ--;

else if(query == '2')

macchiatoQ--;

else if(query == '3')

hotChocolateQ--;

else if(query == '4')

cappuccinoQ--;

else {

cout << "There has been an error. Re-compile program. " << endl;

exit(-1);

}

}

// Function to show the current money status in the machine

// returns: nothing

void CoffeeMachine::showMachineMoney()

{

cout << "\nShowing the current coin status in the machine ->" << endl;

cout << "-> Number of 0.5KM coins: " << numberOfHalfs << endl;

cout << "-> Number of 1KM coins: " << numberOfMarks << endl;

cout << "-> Number of 2KM coins: " << numberOfTwoMarks << endl;

cout << "-> Number of 5KM coins: " << numberOfFiveMarks << endl;

}

// Function to display the error message when machine does not have money to return

// returns: nothing

void CoffeeMachine::insufficient(double money)

{

cout << "\nInsufficient funds." << endl;

cout << "Re-compile program. " << endl;

cout << "Dispensing your " << money << " KM. " << endl;

exit(-1);

}

// Function that controls the user input for further program use.

// returns: false if user enters y, true if user enters n

bool CoffeeMachine::askToContinue(char query)

{

bool controller;

if((query == 'y') || (query == 'Y')) {

controller = false;

}

else if((query == 'n') || (query == 'N')) {

controller = true;

}

return controller;

}

// Function to check if quantities of each member variable is above 0. If obtained value is 0, program ends.

// returns: nothing

void CoffeeMachine::checkQuantity()

{

if((espressoQ == 0) || (macchiatoQ == 0) || (hotChocolateQ == 0) || (cappuccinoQ == 0)) {

cout << "\nThe product you asked for is currently unavailable. \nRe-compile program to try something else. " << endl;

exit(-1);

}

}