Main.cpp

/\*

CECS 282 Sec 06 Week 5 Lab 2

Randy Mondragon

Kenry Yu

Emmanuel Rodriguez

Demo at 5:00PM

\*/

#include "Can.h"

#include <iostream>

#include <string>

using namespace std;

int main() {

Can c1 = Can("Peaches", 15);

Can c2 = Can("Peas", 20);

Can c3 = Can("Soup", 24);

Can c4 = Can().mix(c1, c3); // c4 will have a mixture of c1 and c2

Can().pour(c2, c1); // pour c1 into c2. c1 will be empty

Can().stretch(c3, 20);

cout << "C1 - ";

c1.display(); // empty: 0

cout << "C2 - ";

c2.display(); // Peas, Peaches: 35

cout << "C3 - ";

c3.display(); // Soup: 44

cout << "C4 - ";

c4.display(); // Peaches, Soup: 39

return 0;

}Text, chat or text message

Description automatically generated

Can.h

#ifndef CAN\_H

#define CAN\_H

#include <iostream>

using namespace std;

class Can {

private:

string name;

int weight;

public:

Can();

Can(string name, int weight);

string getName();

int getWeight();

void display();

void pour(Can &, Can &);

Can mix(Can, Can);

void stretch(Can &, int);

};

#endif

Can.cpp

#include "Can.h"

#include <iostream>

Can::Can() {

name = "Empty";

weight = 0;

}

Can::Can(string name, int weight) {

this->name = name;

this->weight = weight;

}

string Can::getName() { return name; }

int Can::getWeight() { return weight; }

void Can::display() { std::cout << weight << " ounce can of " << name << endl; }

void Can::pour(Can &can\_1, Can &can\_2) {

can\_1 = Can(can\_1.getName() + " and " + can\_2.getName(),

can\_1.getWeight() + can\_2.getWeight());

can\_2 = Can();

}

void Can::stretch(Can &c, int v) { c = Can(c.getName(), c.getWeight() + v); }

Can Can::mix(Can can\_1, Can can\_2) {

Can new\_can = Can(can\_1.getName() + " and " + can\_2.getName(),

can\_1.getWeight() + can\_2.getWeight());

return new\_can;

}