/\*

\* Vic Cuatico

\* 400507885

\*

\* Lab 5

\* Question 1

\*/

#include <iostream>

#include "NumDays.h"

using namespace std;

int main()

{

//a

NumDays hours1(23);

NumDays hours2(18);

cout << hours1;

cout << hours2;

//b

cout << hours1 - hours2;

cout << hours1 + hours2;

//c

++hours1, ++hours1;

cout << hours1;

//d

hours2--, hours2--, hours2--;

cout << hours2;

}

A black screen with white text

Description automatically generated

NumDays.h

#pragma once

#include <iostream>

using namespace std;

class NumDays {

private:

double hours;

double days;

public:

void SetHours(double h) { hours = h; }

void SetDays(double d) { days = d; }

double GetHours() { return hours; }

double GetDays() { return days; }

NumDays();

NumDays(double h);

NumDays operator+(NumDays RHS);

NumDays operator-(NumDays RHS);

NumDays operator++();

NumDays operator++(int Postincrement);

NumDays operator--();

NumDays operator--(int postdecrement);

};

ostream& operator<<(ostream& out, NumDays obj);

NumDays.cpp  
#include "NumDays.h"  
NumDays::NumDays() { hours = 0, days = 0; }

NumDays::NumDays(double h) { hours = h, days = (hours / 8); }

NumDays NumDays::operator+(NumDays RHS) {

double sum = hours + RHS.GetHours();

NumDays result(sum);

return result;

}

NumDays NumDays::operator-(NumDays RHS) {

double dif = hours - RHS.GetHours();

NumDays result(dif);

return result;

}

NumDays NumDays::operator++() {

hours += 1;

NumDays result(hours);

return result;

}

NumDays NumDays::operator++(int Postincrement) {

NumDays result(hours);

hours += 1;

return result;

}

NumDays NumDays::operator--(){

hours -= 1;

NumDays result(hours);

return result;

}

NumDays NumDays::operator--(int PostDecrement) {

NumDays result(hours);

hours -= 1;

return result;

}

ostream& operator<<(ostream& out, NumDays obj) {

double Hours = obj.GetHours();

double OutHours = 0;

double Days = obj.GetDays();

double OutDays = 0;

while (Hours >= 8) {

OutDays += 1;

Hours -= 8;

}

OutHours = Hours;

out << OutDays << " Days and " << OutHours << "hours." << endl;

return out;

}

/\*

\* Vic Cuatico

\* 400507885

\*

\* Lab 5

\* Question 2

\*/

#include <iostream>

#include "Dog.h"

using namespace std;

int main()

{

Dog d1("Golden Retriever", 1, "Male", "Adult");

Dog d2("Labrador Retriever", 1, "Female", "Adult");

Dog d3("German Shepard", 1, "Fixed", "Adult");

Dog d4("German Shepard", 1, "Fixed", "Adult");

Dog d5 = d1 + d2;

Dog d6 = d3 + d4;

cout << d5 << endl;

cout << d6 << endl;

}

Dog.h  
#pragma once

#include <iostream>

using namespace std;

class Dog

{

private:

string breed;

int age;

string sex;

string maturity;

public:

Dog();

Dog(string b, int a, string s, string m);

void setS(string s);

string getB();

int getA();

string getS();

string getM();

void fixDog(Dog x);

Dog operator+(Dog x);

};

ostream& operator<<(ostream& out, Dog obj);

Dog.cpp

#include "Dog.h"

Dog::Dog() { breed = "", age = 0, sex = "", maturity = ""; }

Dog::Dog(string b, int a, string s, string m) { breed = b, age = a, sex = s, maturity = m; }

void Dog::setS(string s) { sex = s; }

string Dog::getB() { return breed; }

int Dog::getA() { return age; }

string Dog::getS() { return sex; }

string Dog::getM() { return maturity; }

void Dog::fixDog(Dog x) { x.setS("Fixed"); }

Dog Dog::operator+(Dog x) {

if (x.getS() == "Fixed" or sex == "Fixed" or x.getS() == sex ) {

Dog baby;

return baby;

}

else if (breed == x.getB()) {

Dog baby(breed, 0, "Female", "Puppy");

return baby;

}

else {

Dog baby;

Dog baby("Cross", 0, "Female", "Puppy");

}

}

ostream& operator<<(ostream& out, Dog obj) {

out << "NEW DOG" << endl;

out << "Breed: " << obj.getB() << endl;

out << "Age: " << obj.getA() << endl;

out << "Sex: " << obj.getS() << endl;

out << "Maturity: " << obj.getM() << endl;

return out;

}