/\*

Lab 8

CIS 200 Fall 2014

\*/

#include <iostream>

#include <string>

using namespace std;

const int MAX\_SIDES = 100;

template <class T>

T max(T a, T b)

{

if (a > b) return a;

return b;

}

class Polygon

{

public:

Polygon(int numSides = 0, string n=""): numOfSides(numSides), name(n){ }

void setName(string s){ name = s; }

string getName() { return name; }

void set(int sideNum, int value)

{

if (sideNum < 0 || sideNum > numOfSides || value < 0)

return;

else

sides[sideNum] = value;

}

int get(int sideNum)

{

if (sideNum < 0 || sideNum > 99)

return 0;

else

return sides[sideNum];

}

virtual int perimeter() { return 0; }

virtual double area() { return 0; }

virtual double volume() { return 0; }

private:

int numOfSides;

int sides[MAX\_SIDES];

string name;

};

class Rectangle : public Polygon

{

public:

Rectangle() : Polygon(4, "rectangle") {}

double area() { return get(0) \* get(1); }

int perimeter()

{

return 2 \* (get(0) + get(1));

}

};

/\*

You can write any thing you want in here and the computer will ignore it.

Bruce Elenbogen

Purpose: This is my first c++ program. It will say my name

May 6th, 2015

Elenbogen

\*/

#include<iostream> // this is library of reading and writing routines

#include <fstream>

#include<string>

using namespace std; // includes the console input and output

struct Date{

int day;

int month;

int year;

Date(int d = 0, int m = 0, int y = 0){

day = d;

month = m;

year = y;

}

friend ostream & operator<<(ostream & out, const Date & date){

cout << date.day << "/" << date.month << "/" << date.year;

return out;

}

};

template<class T>

T maxFunc(T array[], int size){

T maxSoFar = array[0];

for (int i = 1; i < size; i++)

if (maxSoFar < array[i])

maxSoFar = array[i];

return maxSoFar;

}

template<class T>

class A

{

T valuea;

public:

A() {}

T getValuea() const { return valuea; }

void setValuea(T x) { valuea = x; }

A(const A & rhs) { valuea = rhs.valuea; }

};

template<class T>

class B : public A<T>

{

T valueb;

public:

B(){}

T getValueb() const { return valueb; }

void setValueb(T const x) { valueb = x; }

B(const B & rhs) {

valueb = rhs.getValueb();

setValuea(rhs.getValuea());

}

B(T valA, T valB){

valueb = valB;

A<T>::setValuea(valA);

}

friend

ostream & operator<<(ostream & out, const B & b){

out << "Value A: " << b.getValuea()

<< " Value B: " << b.getValueb();

return out;

}

};

int main()

{

//

// Question 1

//

int nums[] = { 4, 1, 13, 3, 2 };

double values[] = { 1.1, 4.1, 8.1, 5.2, 2.3 };

string words[] = { "the", "student", "is", "in", "class" };

int size = 5;

cout << "The max of the integers is : " << maxFunc(nums, 5) << endl;

cout << "The max of the doubles is : " << maxFunc(values, 5) << endl;

cout << "The max of the words is : " << maxFunc(words, 5) << endl;

//

// Question 2

//

Date now(27, 10, 2014);

Date later(2, 11, 2014);

B<double> floatB(1.34, 3.14);

B<int> intB(1, 3);

B<char> charB('a', 'b');

B<string> strB("good", "morning");

B<Date> dateB(now, later);

cout << "Question 2 \n";

cout << "floatB is " << floatB << endl;

cout << "intB is " << intB << endl;

cout << "charB is " << charB << endl;

cout << "strB is " << strB << endl;

cout << "dateB is " << dateB << endl;

return 0;

}

