/// Andrew Souza

/// Comp 200 -- Spring 2023

/// Exam 3

#include <iostream>

#include <vector>

using namespace std;

// Implement a base class Shape and a derived classes Rectangle and Triangle.

class Shape {

public:

Shape() {

this->height = 0.00;

this->width = 0.00;

}

Shape(double height, double width) {

this->height = height;

this->width = width;

}

void SetHeight(double height) {

this->height = height;

}

void SetWidth(double width) {

this->width = width;

}

double GetHeight() const {

return this->height;

}

double GetWidth() const {

return this->width;

}

virtual double GetArea() const = 0; // Supply virtual functions double area().

private:

double height;

double width;

};

class Rectangle : public Shape {

public:

Rectangle() {

Shape::SetHeight(0.00);

Shape::SetWidth(0.00);

}

Rectangle(double height, double width) {

Shape::SetHeight(height);

Shape::SetWidth(width);

}

virtual double GetArea() const override {

return (Shape::GetWidth() \* Shape::GetHeight());

}

};

class Triangle : public Shape {

public:

Triangle() {

Shape::SetHeight(0.00);

Shape::SetWidth(0.00);

}

Triangle(double height, double width) {

Shape::SetHeight(height);

Shape::SetWidth(width);

}

virtual double GetArea() const override {

return ((Shape::GetWidth() \* Shape::GetHeight()) / 2);

}

};

int main() {

vector<Shape\*> house;

double totalArea = 0.00;

/\* Fill a vector of Shape\* pointers with a mixture of the shapes

with each shape representing one room within a house. \*/

Rectangle\* livingRoom = new Rectangle(8.04, 16.36);

house.push\_back(livingRoom);

Triangle\* entryWay = new Triangle;

entryWay->SetWidth(10);

entryWay->SetHeight(12);

house.push\_back(entryWay);

Rectangle\* masterBedroom = new Rectangle;

masterBedroom->SetHeight(9.06);

masterBedroom->SetWidth(20.04);

house.push\_back(masterBedroom);

Rectangle\* secondBedroom = new Rectangle(8.07, 14.36);

house.push\_back(secondBedroom);

Triangle\* bathroom = new Triangle(7.75, 5.73);

house.push\_back(bathroom);

Rectangle\* kitchen = new Rectangle;

kitchen->SetWidth(16);

kitchen->SetHeight(9.90);

house.push\_back(kitchen);

/\* Compute the total area of the home by summing all the areas of the shapes within the vector.

Display the total. \*/

for (unsigned int i = 0; i < house.size(); i++) {

totalArea += house.at(i)->GetArea();

}

cout << "Total area of house: " << totalArea << "sqft" << endl;

return 0;

}

