/// Andrew Souza

/// Comp 200 - Spring 23

/// Exam 2

#include <iostream>

using namespace std;

// Question 1

// Create a circle class with the following member functions:

class Circle {

public:

Circle();

Circle(double rad);

double getArea() const;

double getRadius() const;

void setRadius(double rad);

private:

double radius;

};

Circle::Circle() {

radius = 0;

}

Circle::Circle(double rad) {

radius = rad;

}

double Circle::getArea() const {

return (2 \* 3.14 \* radius \* radius);

}

double Circle::getRadius() const {

return radius;

}

void Circle::setRadius(double rad) {

radius = rad;

}

int main() {

Circle myCircle;

Circle newCircle(7);

cout << "Radius of myCircle: " << myCircle.getRadius() << endl;

myCircle.setRadius(10);

cout << "New Radius of myCircle: " << myCircle.getRadius() << endl;

cout << "Area of myCirlce: " << myCircle.getArea() << endl;

cout << "Radius of newCircle: " << newCircle.getRadius() << endl;

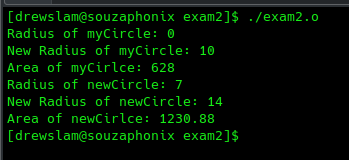
newCircle.setRadius(newCircle.getRadius() + 7);

cout << "New Radius of newCircle: " << newCircle.getRadius() << endl;

cout << "Area of newCirlce: " << newCircle.getArea() << endl;

return 0;

}



// Question 2

/\* Implement the relational operators (<, <=, ==, !=, >, >=) in a circle class.

\* The relational operators are used to compare

\* to the radii of the circle.\*/

bool operator<(const Circle& l, const Circle& r) {

return l.getRadius() < r.getRadius();

}

bool operator<=(const Circle& l, const Circle& r) {

return l.getRadius() <= r.getRadius();

}

bool operator==(const Circle& l, const Circle& r) {

return l.getRadius() == r.getRadius();

}

bool operator!=(const Circle& l, const Circle& r) {

return l.getRadius() != r.getRadius();

}

bool operator>(const Circle& l, const Circle& r) {

return l.getRadius() > r.getRadius();

}

bool operator>=(const Circle& l, const Circle& r) {

return l.getRadius() >= r.getRadius();

}

int main() {

Circle circle1(7.5);

Circle circle2(8.25);

Circle circle3(3.1);

Circle circle4(10);

Circle circle5(15.76);

Circle circle6(3.1);

cout << "Circle 1 > Circle 2: " << (circle1 > circle2) << endl;

cout << "Circle 2 < Circle 3: " << (circle2 < circle3) << endl;

cout << "Circle 3 == Circle 4: " << (circle3 == circle4) << endl;

cout << "Circle 6 >= Circle 5: " << (circle6 >= circle5) << endl;

cout << "Circle 5 >= Circle 4: " << (circle5 >= circle4) << endl;

cout << "Circle 4 != Circle 3: " << (circle4 != circle3) << endl;

cout << "Circle 3 == Circle 6: " << (circle3 == circle6) << endl;

return 0;

}

