[ Ravi Patel ] Instructor: Dr. Hindo

CPSC 230

Chapter 10 - lab assignment 10-2 (15 pts.)

Drop in assignment 10 dropbox

Q1(10 points)

Write RegularPolygon class that define the followings:

1. Set the private members: the “number\_side” as int, the “apothem” and “sideLength” as double,
2. 2- Define the constructor:

RegularPolygon (double side\_length, int number\_sides); //set the apothem, sideLength and the number of sides.

3- Define the constructor:   
 RegularPolygon ( );// initialize the sideLength = 1 and number\_sides=4

4- public function to find the apothem:

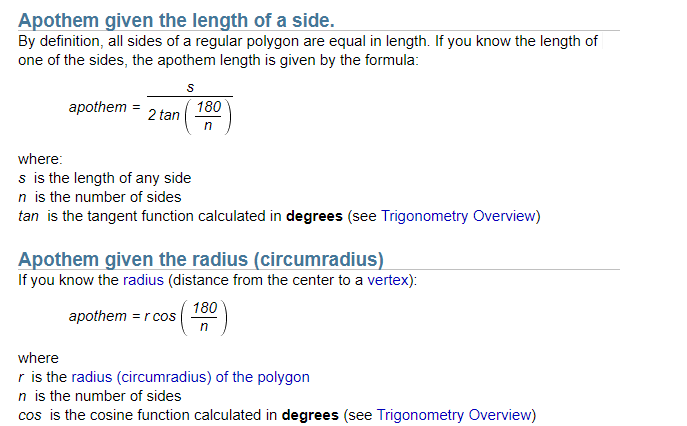
double apothem ( ); // find and return the apothem

5- public function to find the area: double area( )

6- public function to find the perimeter: double perimeter ()

**Note**: **Area** of a **Regular Polygon**. The **area** of a **regular polygon** is given by the formula below. **area** = (½)(apothem)(perimeter)

Note: Find the



#include <iostream>

#include <math.h>

using namespace std;

class RegularPolygon{

private:

double number\_sides; //declare n\_s

double apothem, sideLength; //declare a, sL

public:

double get\_apothem(); //find and return the apothem

double get\_area(); //a=0.5\*apothem\*perimeter

double get\_perimeter();

RegularPolygon(){

sideLength = 1; //initialize sL = 1

number\_sides = 4; //initialize n\_s = 4

}

RegularPolygon(double side\_Length, double number\_side){

number\_sides = number\_side;

sideLength = side\_Length;

}

};

int main(int argc, char \*argv[]) {

RegularPolygon square(1,4);

RegularPolygon sq;

cout<<"Apothem: "<<sq.get\_apothem()<<endl;

cout<<"Area: "<<sq.get\_area()<<endl;

cout<<"Perimeter: "<<sq.get\_perimeter()<<endl;

cout<<endl;

cout<<"Apothem: "<<square.get\_apothem()<<endl;

cout<<"Area: "<<square.get\_area()<<endl;

cout<<"Perimeter: "<<square.get\_perimeter()<<endl;

return 0;

}

double RegularPolygon::get\_apothem(){

return ((sideLength/2) \* cos((180/number\_sides)));

}

double RegularPolygon::get\_perimeter(){

return (sideLength \* number\_sides);

}

double RegularPolygon::get\_area(){

return ((RegularPolygon::get\_perimeter() \* apothem)/2);

}

**SAMPLE OUTPUT:**

**Apothem: 0.262661**

**Area: 1.39062e-309**

**Perimeter: 4**

**Apothem: 0.262661**

**Area: 1.39062e-309**

**Perimeter: 4**

Q2 (5 points)-

Complete the missing statements in slide 39 (simple inheritance example). Can base class access derived class? **(Yes)** Can derived class access base class? **(No, error)**

#include <iostream>

class BaseClass {

public:

BaseClass( ) : baseValue (20) { };

BaseClass(int aValue) : baseValue (aValue) { };

int baseValue;

};

class DerivedClass : public BaseClass {

public:

DerivedClass() : derivedValue (10) { }; // a short way to write a constructor

DerivedClass (int aDerivedValue) : BaseClass(15), derivedValue(aDerivedValue){ }; //// a short way to write a constructor

int getDerivedValue( ) { return derivedValue; }

private:

int derivedValue;

};

int main( )

{

BaseClass base;

DerivedClass child(5);

std::cout<<"baseValue using base object: "<<base.baseValue<<std::endl;

std::cout<<"baseValue using child object: "<<child.baseValue<<std::endl;

std::cout<<"derivedValue using child object: "<<child.getDerivedValue()<<std::endl;

BaseClass Base(100);

std::cout<<"baseValue using BaseClass: "<<Base.baseValue<<std::endl;

std::cout<<"Impossible to print bassValue using Derived Class! Error.";

return 0;

}

**SAMPLE OUTPUT:**

**baseValue using base object: 20**

**baseValue using child object: 15**

**derivedValue using child object: 5**

**baseValue using BaseClass: 100**

**Impossible to print bassValue using Derived Class! Error.**