Exam 1

In your own words, show your work.

For this exam, you will write a program which will satisfy the various elements. This program is to request the measure for the side of a rectangular, the diameter of a circular, and the base of a regular hexagon pizza. It is then to compute the area of each figure and determine their ranking from small to medium to large (an enumerated type). Store this value as a private variable, whole, of the enumerated type. Then you are to determine the area of slices with a square having nine, a circle having eight, and a hexagon having 6. Finally you are to rank the slice areas in a manner similar to the whole pizza. Store this value as a private variable, slice, of the enumerated type. Open the file “comparisons.prn” to receive the output of your program. The output should include the following for each pizza:

1. The area of the whole pizza,
2. Its whole area rank,
3. The area of its slices, and
4. The rank of its slice area.

Adhere to the commenting guide for this program. Test your program for pizzas of the following sizes:

1. Square: 12 inches
2. Circle: 14 inches
3. Hexagon: 8 inches.

Submit this document and all of the program files (header, source, and output) in a single zip file.

1. Insert here the content of the header file for the enumerated type which identifies the ranks: small, medium, and large.

//SIZE\_H.h

//Author: Erick Narvaez

//Classes/Procedures: (none)

#ifndef SIZE\_H

#define SIZE\_H

//Included here because each file using this enum will need to output the string value, not the integer, in its toString method

#include <string>

//Size will be used to compare different pizza classes using a standard size

enum Size {

SMALL = 0,

MEDIUM,

LARGE

};

#endif

1. Insert here the content of the header file for the square class.

//SQUARE\_H

// Author: Erick Narvaez

// Classes/Procedure: Square Class

#ifndef SQUARE\_H

#define SQUARE\_H

#include "SIZE\_H.h"

// Class: Square

// Author: Erick Narvaez

// Summary: The square class represents the square pizza. It takes in the side length provided by the user and calculates its area, slice area

// and determines whether these sizes fall under a small, medium or large category. It also has a toString method to output these values

// in a formated string.

class Square {

private:

Size whole;

Size slice;

double sideLength;

double area;

double sliceArea;

public:

Square(double length);

void calcArea();

void setWholeSize(Size size);

void setSliceSize(Size size);

void calcSliceArea();

double getArea();

double getSliceArea();

std::string toString();

std::string size\_Name(int size);

};

#endif

1. Insert here the content of the header file for the circle class.

//CIRCLE\_H

// Author: Erick Narvaez

// Classes/Procedure: Circle Class

#ifndef CIRCLE\_H

#define CIRCLE\_H

#include "SIZE\_H.h"

// Class: Circle

// Author: Erick Narvaez

// Summary: The circle class represents the circle pizza. It takes in the diameter provided by the user and calculates its area, slice area

// and determines whether these sizes fall under a small, medium or large category. It also has a toString method to output these values

// in a formated string.

class Circle {

private:

Size whole;

Size slice;

double diameter;

double area;

double sliceArea;

public:

Circle(double sideLength);

double getArea();

double getSliceArea();

void setWholeSize(Size size);

void setSliceSize(Size size);

void calcArea();

void calcSliceArea();

std::string toString();

std::string size\_Name(int size);

};

#endif

1. Insert here the content of the header file for the hexagon class.

//HEXAGON\_H

// Author: Erick Narvaez

// Classes/Procedure: Hexagon Class

#ifndef HEXAGON\_H

#define HEXAGON\_H

#include "SIZE\_H.h"

// Class: Hexagon

// Author: Erick Narvaez

// Summary: The hexagon class represents the hexagon pizza. It takes in the side length provided by the user and calculates its area, slice area

// and determines whether these sizes fall under a small, medium or large category. It also has a toString method to output these values

// in a formated string.

class Hexagon {

private:

Size whole;

Size slice;

double sideLength;

double area;

double sliceArea;

public:

Hexagon(double sideLength);

void calcArea();

void calcSliceArea();

void setWholeSize(Size size);

void setSliceSize(Size size);

double getArea();

double getSliceArea();

std::string toString();

std::string size\_Name(int size);

};

#endif

1. Insert here the content of the class file for the square class.

//Square.cpp

// Author: Erick Narvaez

// Class/Procedures:

// 1) Square(): Constructor for Square class. Accepts a side length and calculates the rest of the properties

// 2) calcArea(): Will calculate the area using the sideLength property and sets the area property of the object

// 3) setWholeSize(): Will set the enum whole size property using the area property

// 4) setSliceSize(): Will set the enum slice size property using the slice area property

// 5) calcSliceArea(): Takes the whole area and calculates the slice area

// 6) toString(): Returns the properties of Square in a formatted string

// 7) size\_Name(): Converts Size enum to string name

// 8) getArea(): Returns the area of the object

// 9) getSliceArea(): Returns the slice area of the object

#include "SQUARE\_H.h"

#include <iostream>

// Name: Square()

// Author: Erick Narvaez

// Summary: Contructor for the Square class. Sets the value of the following properties for the

// object: sideLength, area, slice area, and size enums.

// Arguments:

// 1) double length (input). The side length of the square object.

Square::Square(double length) {

sideLength = length;

whole = MEDIUM;

slice = MEDIUM;

calcArea();

calcSliceArea();

//setWholeSize();

//setSliceSize();

}

// Name: calcArea()

// Author: Erick Narvaez

// Summary: Calculates the area of the square by multiplying the length and width. (A=l\*w)

// Arguments: (none)

void Square::calcArea() {

area = sideLength\*sideLength;

}

// Name: setWholeSize()

// Author: Erick Narvaez

// Summary: Determines what size the pizza falls under. Assumes that a small pizza has an area less than 36 inches^2,

// a medium has between 36 and 144, and a large has 144 or more.

// Arguments: (none)

void Square::setWholeSize(Size size) {

whole = size;

/\*if (area <= 36)

whole = SMALL;

else if (area >= 144)

whole = LARGE;

else

whole = MEDIUM;\*/

}

// Name: setSliceSize()

// Author: Erick Narvaez

// Summary: Determines what size the slice of the pizza falls under. Assumes that a small slice of pizza has an area less than 4 inches^2,

// a medium has between 4 and 16, and a large has 16 or more.

// Arguments: (none)

void Square::setSliceSize(Size size) {

slice = size;

/\*if (sliceArea <= 4)

slice = SMALL;

else if (sliceArea >= 16)

slice = LARGE;

else

slice = MEDIUM;\*/

}

// Name: calcSliceArea()

// Author: Erick Narvaez

// Summary: Calculates the area of a slice of pizza using the area property of the object.

// Arguments: (none)

void Square::calcSliceArea() {

sliceArea = area / 9;

}

// Name: toString()

// Author: Erick Narvaez

// Summary: Takes all properties of the object and outputs them with identifying strings for the output file in the main function.

// Arguments: (none)

std::string Square::toString() {

std::string str = "";

str += "Type of Pizza: Square\n";

str += "Area of Whole Pizza: " + std::to\_string(area) + "\n";

str += "Whole Area Rank: " + size\_Name(whole) + "\n";

str += "Area of Slice: " + std::to\_string(sliceArea) + "\n";

str += "Slice Area Rank: " + size\_Name(slice) + "\n";

return str;

}

// Name: size\_Name()

// Author: Erick Narvaez

// Summary: Determines and returns what the String name for the Size enum value is.

// Arguments: (none)

std::string Square::size\_Name(int size) {

switch (size) {

case(0) : return "Small";

case(1) : return "Medium";

case(2) : return "Large";

default: return "not a size";

}

}

// Name: getArea()

// Author: Erick Narvaez

// Summary: Getter method for the area of the object. Returns a double.

// Arguments: (none)

double Square::getArea() {

return area;

}

// Name: getArea()

// Author: Erick Narvaez

// Summary: Getter method for the slice area of the object. Returns a double.

// Arguments: (none)

double Square::getSliceArea() {

return sliceArea;

}

1. Insert here the content of the class file for the circle class.

//Circle.cpp

// Author: Erick Narvaez

// Class/Procedures:

// 1) Circle(): Constructor for Circle class. Accepts a diameter length and calculates the rest of the properties

// 2) calcArea(): Will calculate the area using the sideLength property and sets the area property of the object

// 3) setWholeSize(): Will set the enum whole size property using the area property

// 4) setSliceSize(): Will set the enum slice size property using the slice area property

// 5) calcSliceArea(): Takes the whole area and calculates the slice area

// 6) toString(): Returns the properties of Square in a formatted string

// 7) size\_Name(): Converts Size enum to string name

// 8) getArea(): Returns the area of the object

// 9) getSliceArea(): Returns the slice area of the object

#include "CIRCLE\_H.h"

#include <iostream>

#include <math.h>

// Name: Circle()

// Author: Erick Narvaez

// Summary: Contructor for the Circle class. Sets the value of the following properties for the

// object: diameter, area, slice area, and size enums.

// Arguments:

// 1) double length (input). The diameter length of the circle object.

Circle::Circle(double length) {

diameter = length;

whole = MEDIUM;

slice = MEDIUM;

calcArea();

calcSliceArea();

}

// Name: calcArea()

// Author: Erick Narvaez

// Summary: Calculates the area of the circle by multiplying pi by the radius squared. (A=pi\*r^2)

// Arguments: (none)

void Circle::calcArea() {

double pi = 3.1415926535897;

area = pow(diameter/2,2)\*pi;

}

// Name: setWholeSize()

// Author: Erick Narvaez

// Summary: Determines what size the pizza falls under. Assumes that a small pizza has an area less than 36 inches^2,

// a medium has between 36 and 144, and a large has 144 or more.

// Arguments: (none)

void Circle::setWholeSize(Size size) {

whole = size;

}

// Name: setSliceSize()

// Author: Erick Narvaez

// Summary: Determines what size the slice of the pizza falls under. Assumes that a small slice of pizza has an area less than 4 inches^2,

// a medium has between 4 and 16, and a large has 16 or more.

// Arguments: (none)

void Circle::setSliceSize(Size size) {

slice = size;

}

// Name: calcSliceArea()

// Author: Erick Narvaez

// Summary: Calculates the area of a slice of pizza using the area property of the object.

// Arguments: (none)

void Circle::calcSliceArea() {

sliceArea = area / 8;

}

// Name: toString()

// Author: Erick Narvaez

// Summary: Takes all properties of the object and outputs them with identifying strings for the output file in the main function.

// Arguments: (none)

std::string Circle::toString() {

std::string str = "";

str += "Type of Pizza: Circle\n";

str += "Area of Whole Pizza: " + std::to\_string(area) + "\n";

str += "Whole Area Rank: " + size\_Name(whole) + "\n";

str += "Area of Slice: " + std::to\_string(sliceArea) + "\n";

str += "Slice Area Rank: " + size\_Name(slice) + "\n";

//std::cout << str;

return str;

}

// Name: size\_Name()

// Author: Erick Narvaez

// Summary: Determines and returns what the String name for the Size enum value is.

// Arguments: (none)

std::string Circle::size\_Name(int size) {

switch (size) {

case(0) : return "Small";

case(1) : return "Medium";

case(2) : return "Large";

default: return "not a size";

}

}

// Name: getArea()

// Author: Erick Narvaez

// Summary: Getter method for the area of the object. Returns a double.

// Arguments: (none)

double Circle::getArea() {

return area;

}

// Name: getArea()

// Author: Erick Narvaez

// Summary: Getter method for the slice area of the object. Returns a double.

// Arguments: (none)

double Circle::getSliceArea() {

return sliceArea;

}

1. Insert here the content of the class file for the hexagon class.

//Hexagon.cpp

// Author: Erick Narvaez

// Class/Procedures:

// 1) Hexagon(): Constructor for Square class. Accepts a side length and calculates the rest of the properties

// 2) calcArea(): Will calculate the area using the sideLength property and sets the area property of the object

// 3) setWholeSize(): Will set the enum whole size property using the area property

// 4) setSliceSize(): Will set the enum slice size property using the slice area property

// 5) calcSliceArea(): Takes the whole area and calculates the slice area

// 6) toString(): Returns the properties of Square in a formatted string

// 7) size\_Name(): Converts Size enum to string name

// 8) getArea(): Returns the area of the object

// 9) getSliceArea(): Returns the slice area of the object

#include "HEXAGON\_H.h"

#include <iostream>

#include <math.h>

// Name: Hexagon()

// Author: Erick Narvaez

// Summary: Contructor for the Hexagon class. Sets the value of the following properties for the

// object: sideLength, area, slice area, and size enums.

// Arguments:

// 1) double length (input). The side length of the hexagon object.

Hexagon::Hexagon(double length) {

sideLength = length;

whole = MEDIUM;

slice = MEDIUM;

calcArea();

calcSliceArea();

}

// Name: calcArea()

// Author: Erick Narvaez

// Summary: Calculates the area of the hexagon by using the hexagon area formula. (A=3\*sqrt(3)\*.5\*s^2)

// Arguments: (none)

void Hexagon::calcArea() {

area = 3\*sqrt(3)\*.5\*pow(sideLength,2);

}

// Name: setWholeSize()

// Author: Erick Narvaez

// Summary: Determines what size the pizza falls under. Assumes that a small pizza has an area less than 36 inches^2,

// a medium has between 36 and 144, and a large has 144 or more.

// Arguments: (none)

void Hexagon::setWholeSize(Size size) {

whole = size;

}

// Name: setSliceSize()

// Author: Erick Narvaez

// Summary: Determines what size the slice of the pizza falls under. Assumes that a small slice of pizza has an area less than 4 inches^2,

// a medium has between 4 and 16, and a large has 16 or more.

// Arguments: (none)

void Hexagon::setSliceSize(Size size) {

slice = size;

}

// Name: calcSliceArea()

// Author: Erick Narvaez

// Summary: Calculates the area of a slice of pizza using the area property of the object.

// Arguments: (none)

void Hexagon::calcSliceArea() {

sliceArea = area / 6;

}

// Name: toString()

// Author: Erick Narvaez

// Summary: Takes all properties of the object and outputs them with identifying strings for the output file in the main function.

// Arguments: (none)

std::string Hexagon::toString() {

std::string str = "";

str += "Type of Pizza: Hexagon\n";

str += "Area of Whole Pizza: " + std::to\_string(area) + "\n";

str += "Whole Area Rank: " + size\_Name(whole) + "\n";

str += "Area of Slice: " + std::to\_string(sliceArea) + "\n";

str += "Slice Area Rank: " + size\_Name(slice) + "\n";

//std::cout << str;

return str;

}

// Name: size\_Name()

// Author: Erick Narvaez

// Summary: Determines and returns what the String name for the Size enum value is.

// Arguments: (none)

std::string Hexagon::size\_Name(int size) {

switch (size) {

case(0) : return "Small";

case(1) : return "Medium";

case(2) : return "Large";

default: return "not a size";

}

}

// Name: getArea()

// Author: Erick Narvaez

// Summary: Getter method for the area of the object. Returns a double.

// Arguments: (none)

double Hexagon::getArea() {

return area;

}

// Name: getArea()

// Author: Erick Narvaez

// Summary: Getter method for the slice area of the object. Returns a double.

// Arguments: (none)

double Hexagon::getSliceArea() {

return sliceArea;

}

1. Insert here the content of the file for the main program code.

// Exam1.cpp

// Author: Erick Narvaez

// Classes/Procedures:

// int main(): Main method; requests user input for shape dimensions and writes specifications to "comparisons.prn".

#include <iostream>

#include <fstream>

#include "SQUARE\_H.h"

#include "CIRCLE\_H.h"

#include "HEXAGON\_H.h"

// Procedure: main()

// Author: Erick Narvaez

// Summary: The main method is the entry point to the program. It requests the side length or diameter for the different types

// of pizzas and then uses this data to create an instance of each pizza. Each pizza class has a toString() method to

// output the required information about each pizza. The main method uses this to output each toString to the "comparison.prn"

// file and to the console.

//Arguments: (none)

int main()

{

using namespace std;

//Get square pizza side length from user

cout << "Side length of square pizza (inches): ";

double squareLength;

cin >> squareLength;

//Get circle pizza diameter from user

cout << "Diameter of circle pizza (inches): ";

double circleDiameter;

cin >> circleDiameter;

//Get hexagon side length from user

cout << "Side length of hexagonal pizza (inches): ";

double hexagonalLength;

cin >> hexagonalLength;

cout << “\n\n”;

//Declare and initialize Square, Circle and Hexagon pizza classes using user input

Square Square1 = Square(squareLength);

Circle Circle1 = Circle(circleDiameter);

Hexagon Hexagon1 = Hexagon(hexagonalLength);

//Compare the areas of the whole between each of the 3 classes

//TODO: Figure out how to compare whole sizes and slice sizes!!

if (Square1.getArea() < Circle1.getArea()) {

if (Square1.getArea() < Hexagon1.getArea()) {

Square1.setWholeSize(SMALL);

if (Hexagon1.getArea() < Circle1.getArea()) {

Circle1.setWholeSize(LARGE);

}

else {

Hexagon1.setWholeSize(LARGE);

}

}

else {

Hexagon1.setWholeSize(SMALL);

Circle1.setWholeSize(LARGE);

}

}

else {

if (Circle1.getArea() < Hexagon1.getArea()) {

Circle1.setWholeSize(SMALL);

if (Hexagon1.getArea() < Square1.getArea()) {

Square1.setWholeSize(LARGE);

}

else {

Hexagon1.setWholeSize(LARGE);

}

}

else {

Hexagon1.setWholeSize(SMALL);

Square1.setWholeSize(LARGE);

}

}

//Compare the areas of the slices between each of the 3 classes

if (Square1.getSliceArea() < Circle1.getSliceArea()) {

if (Square1.getSliceArea() < Hexagon1.getSliceArea()) {

Square1.setSliceSize(SMALL);

if (Hexagon1.getSliceArea() < Circle1.getSliceArea()) {

Circle1.setSliceSize(LARGE);

}

else {

Hexagon1.setSliceSize(LARGE);

}

}

else {

Hexagon1.setSliceSize(SMALL);

Circle1.setSliceSize(LARGE);

}

}

else {

if (Circle1.getSliceArea() < Hexagon1.getSliceArea()) {

Circle1.setSliceSize(SMALL);

if (Hexagon1.getSliceArea() < Square1.getSliceArea()) {

Square1.setSliceSize(LARGE);

}

else {

Hexagon1.setSliceSize(LARGE);

}

}

else {

Hexagon1.setSliceSize(SMALL);

Square1.setSliceSize(LARGE);

}

}

//Output toString methods of each object to the console

cout << Square1.toString() << "\n" << Circle1.toString() << "\n" << Hexagon1.toString();

//Declare output file stream object to open "comparisons.prn" for writing

ofstream outf("comparisons.prn");

//Check if file was successfully opened, if not print error message

if (!outf) {

cerr << "Could not open comparisons.prn for writing.";

exit(1);

}

//Output each toString of each object to the file

outf << Square1.toString() << endl;

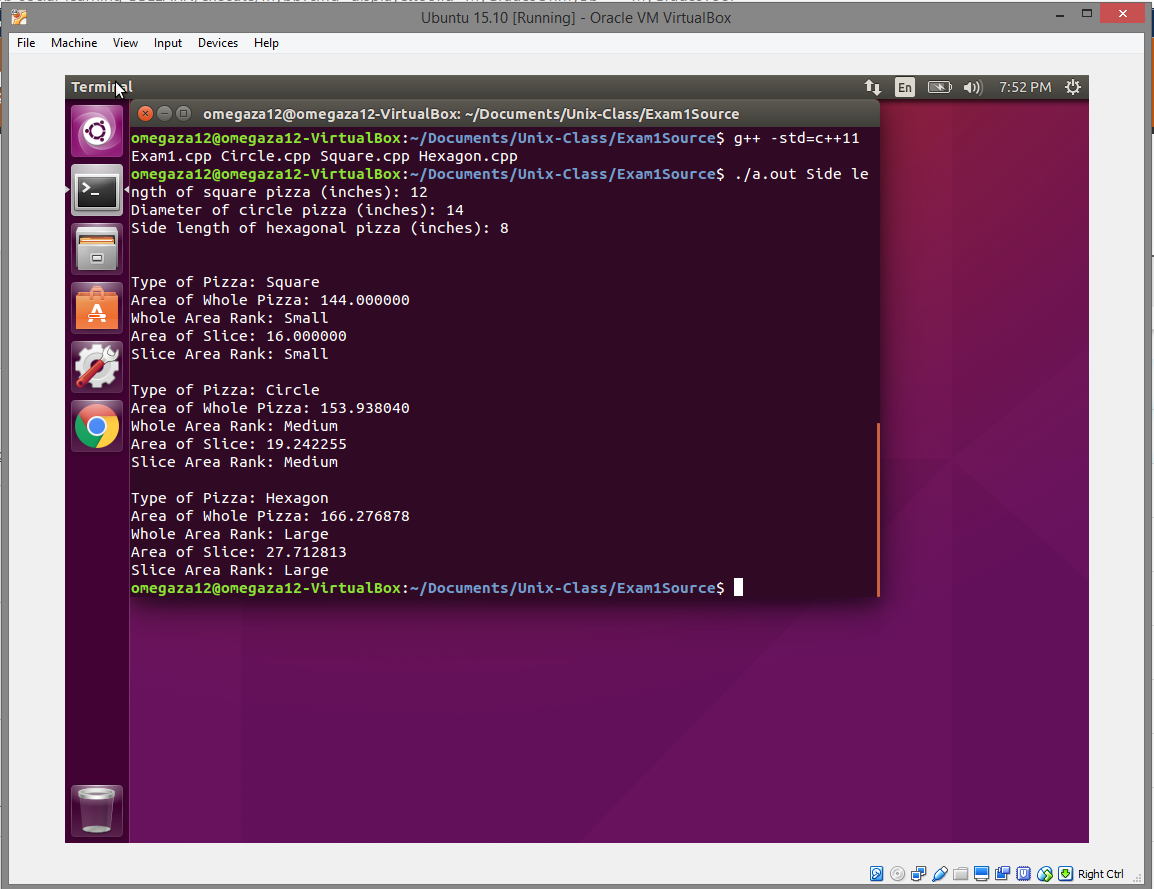
outf << Circle1.toString() << endl;

outf << Hexagon1.toString() << endl;

return 0;

}

1. Insert here a screen shot for the successful compilation and execution of your program within the Ubuntu 15.10 environment.



1. Insert here the content of the output file.

Type of Pizza: Square

Area of Whole Pizza: 144.000000

Whole Area Rank: Small

Area of Slice: 16.000000

Slice Area Rank: Small

Type of Pizza: Circle

Area of Whole Pizza: 153.938040

Whole Area Rank: Medium

Area of Slice: 19.242255

Slice Area Rank: Medium

Type of Pizza: Hexagon

Area of Whole Pizza: 166.276878

Whole Area Rank: Large

Area of Slice: 27.712813

Slice Area Rank: Large