Final Exam Part 1

Source Code

// Preston Knibbe

// Final Exam Part 1

#include <iostream>

using namespace std;

class Animal {

public:

Animal();

Animal(string, string, string);

void setLatinName(string);

void setCommonName(string);

void setSize(string);

string getLatinName();

string getCommonName();

string getSize();

// Prints animal data

void printData();

private:

string latinName;

string commonName;

string Size;

};

Animal::Animal() {

}

Animal::Animal(string latin, string common, string Size2) {

latinName = latin;

commonName = common;

if (Size2 == "Large" || Size2 == "Medium" || Size2 == "Small") {

Size = Size2;

} else {

cout << "Size parameter must be either Large, Medium, or Small" << endl;

}

}

void Animal::setLatinName(string latin) {

latinName = latin;

}

void Animal::setCommonName(string common) {

commonName = common;

}

void Animal::setSize(string Size2) {

if (Size2 == "Large" || Size2 == "Medium" || Size2 == "Small") {

Size = Size2;

} else {

cout << "Size parameter must be either Large, Medium, or Small" << endl;

}

}

string Animal::getLatinName() {

return latinName;

}

string Animal::getCommonName() {

return commonName;

}

string Animal::getSize() {

return Size;

}

void Animal::printData() {

cout << "Common Name: " << commonName << endl;

cout << "Latin Name: " << latinName << endl;

cout << "Size of Animal: " << Size << endl;

cout << "\n";

}

int main()

{

Animal Parakeet("parakeetusmaximus", "Parakeet", "Small");

Parakeet.printData();

Animal Tiger;

Tiger.setLatinName("tigris");

Tiger.setCommonName("Tiger");

Tiger.setSize("Large");

Tiger.printData();

return 0;

}

Code Output

Common Name: Parakeet

Latin Name: parakeetusmaximus

Size of Animal: Small

Common Name: Tiger

Latin Name: tigris

Size of Animal: Large

Process returned 0 (0x0) execution time : 0.041 s

Press any key to continue.

Final Exam Part 2

Source Code

// Preston Knibbe

// Final Exam Part 2

#include <iostream>

#include <math.h>

using namespace std;

class Tube {

public:

Tube();

Tube(double, double, double);

//Prints data about the tube

void displayData();

private:

double innerRadius;

double outerRadius;

double tubeHeight;

//Defines PI for use in program

const double PI = 3.14159;

//Calculates Surface Area

double surfaceArea();

//Calculates Tube Volume

double tubeVolume();

//Calculates Tube Thickness

double tubeThickness();

};

Tube::Tube() {

}

Tube::Tube(double inner, double outer, double height) {

innerRadius = inner;

outerRadius = outer;

tubeHeight = height;

}

void Tube::displayData() {

cout << "Inner Radius: " << innerRadius << endl;

cout << "Outer Radius: " << outerRadius << endl;

cout << "Tube Height: " << tubeHeight << endl;

cout << "Surface Area: " << surfaceArea() << endl;

cout << "Volume: " << tubeVolume() << endl;

cout << "Tube Thickness: " << tubeThickness() << endl;

}

double Tube::surfaceArea() {

double p1 = 2 \* PI \* ((outerRadius\*outerRadius)-(innerRadius\*innerRadius));

double p2 = (2\*PI\*outerRadius\*tubeHeight) + (2\*PI\*innerRadius\*tubeHeight);

return p1 + p2;

}

double Tube::tubeVolume() {

return PI \* ((outerRadius\*outerRadius) - (innerRadius\*innerRadius)) \* tubeHeight;

}

double Tube::tubeThickness() {

return outerRadius - innerRadius;

}

int main()

{

Tube Tube1(20, 30, 500);

Tube1.displayData();

return 0;

}

Code Output

Inner Radius: 1

Outer Radius: 2

Tube Height: 5

Surface Area: 113.097

Volume: 47.1238

Tube Thickness: 1

Process returned 0 (0x0) execution time : 0.009 s

Press any key to continue.

Inner Radius: 20

Outer Radius: 30

Tube Height: 500

Surface Area: 160221

Volume: 785398

Tube Thickness: 10

Process returned 0 (0x0) execution time : 0.043 s

Press any key to continue.