Homework 6

Source Code

Cube

// Name: Preston Knibbe

//

// Shape: Cube

//

// Equation:

// Surface Area = 6 \* (side)^2

// Volume = (side)^3

#include <cmath>

#include <iostream>

using std::cout;

using std::endl;

using std::fixed;

#include <iomanip>

using std::setw;

using std::setprecision;

using std::showpoint;

using std::setfill;

#define PI 3.14159265358979323846

// Fully implement the following class

class Cube

{

public:

Cube(double = 1); // double is the Side

void SetCube(double); // double is the Side,

// Should set volume, and area

// too.

double GetSide(); // Return the Side

double SurfaceArea(); // Calculates the SufaceArea

double Volume(); // Calculates the Volume

void PrintCube(); // Prints Information about

// the Cube

void SideFromArea(double); // Calculate the Side of

// of a Cube given its Area

void SideFromVolume(double); // Calculate the Side of

// of a Cube given its Volume

private:

double side;

double volume;

double surfacearea;

};

Cube::Cube(double s)

{

SetCube(s);

}

void Cube::SetCube(double s)

{

side = s;

surfacearea = 6 \* s \* s;

volume = s \* s \* s;

}

double Cube::GetSide()

{

return side;

}

double Cube::SurfaceArea()

{

return surfacearea;

}

double Cube::Volume()

{

return volume;

}

void Cube::PrintCube()

{

cout << "Side: " << side << endl;

cout << "Volume: " << volume << endl;

cout << "Surface Area: " << surfacearea << endl;

}

void Cube::SideFromArea(double area)

{

double new\_side = sqrt(area/6);

SetCube(new\_side);

}

void Cube::SideFromVolume(double volume)

{

double new\_side = pow(volume/6, 1.0/3.0);

SetCube(new\_side);

}

main()

{

Cube C1(2), C2, C3(5), C4, C5;

C1.PrintCube();

C2.PrintCube();

C2.SideFromVolume(250);

C2.PrintCube();

C3.PrintCube();

C4.SetCube(10);

C4.PrintCube();

C5.SideFromArea(30);

C5.PrintCube();

}

/////////////////////////////////////////////////////////////////////////////

Cylinder

// Name: Preston Knibbe

//

// Shape: Cylinder

//

// Equation:

// Surface Area = 2 \* PI \* r \* (r+h)

// Volume = PI \* r^2 \* h

#include <cmath>

#include <iostream>

using std::cout;

using std::endl;

using std::fixed;

#include <iomanip>

using std::setw;

using std::setprecision;

using std::showpoint;

using std::setfill;

#define PI 3.14159265358979323846

// Fully implement the following class

class Cylinder

{

public:

Cylinder(double = 1, double = 1); // double is the Side

void SetCylinder(double, double); // double is the Side,

// Should set volume, and area

// too.

double GetRadius(); // Return the Radius

double GetHeight(); // Return the Height

double SurfaceArea(); // Calculates the SufaceArea

double Volume(); // Calculates the Volume

void PrintCylinder(); // Prints Information about

// the Cylinder

void HeightFromAreaAndRadius(double, double); // Calculate the Height of

// of a Cylinder given its Area and Radius

void HeightFromVolumeAndRadius(double, double); // Calculate the Height of

// of a Cylinder given its Volume and Radius

private:

double radius;

double height;

double volume;

double surfacearea;

};

Cylinder::Cylinder(double r, double h)

{

SetCylinder(r, h);

}

void Cylinder::SetCylinder(double r, double h)

{

radius = r;

height = h;

surfacearea = 2\*PI\*r\*(r+h);

volume = PI\*r\*r\*h;

}

double Cylinder::GetRadius()

{

return radius;

}

double Cylinder::GetHeight() {

return height;

}

double Cylinder::SurfaceArea()

{

return surfacearea;

}

double Cylinder::Volume()

{

return volume;

}

void Cylinder::PrintCylinder()

{

cout << "Radius: " << radius << endl;

cout << "Height: " << height << endl;

cout << "Volume: " << volume << endl;

cout << "Surface Area: " << surfacearea << endl;

}

void Cylinder::HeightFromAreaAndRadius(double area, double radius)

{

double new\_height = area/(2\*PI\*radius) - radius;

SetCylinder(radius, new\_height);

}

void Cylinder::HeightFromVolumeAndRadius(double volume, double radius)

{

double new\_height = volume/(PI \* radius \* radius);

SetCylinder(radius, new\_height);

}

main()

{

Cylinder C1(2, 5), C2, C3(5, 10), C4, C5;

C1.PrintCylinder();

C2.PrintCylinder();

C2.HeightFromVolumeAndRadius(250, 10);

C2.PrintCylinder();

C3.PrintCylinder();

C4.SetCylinder(10, 20);

C4.PrintCylinder();

C5.HeightFromAreaAndRadius(500, 5);

C5.PrintCylinder();

}

/////////////////////////////////////////////////////////////////////////////

Torus

// Name: Preston Knibbe

//

// Shape: Torus

//

// Equation:

// Surface Area = 4 \* PI^2 \* R \* r

// Volume = 2 \* PI^2 \* R \* r^2

#include <cmath>

#include <iostream>

using std::cout;

using std::endl;

using std::fixed;

#include <iomanip>

using std::setw;

using std::setprecision;

using std::showpoint;

using std::setfill;

#define PI 3.14159265358979323846

// Fully implement the following class

class Torus

{

public:

Torus(double = 1, double = 1); // doubles are the BigR and r

void SetTorus(double, double); // doubles are the BigR and r,

// Should set volume, and area

// too.

double GetBigRadius(); // Return the Radius

double GetLittleRadius(); // Return the

double SurfaceArea(); // Calculates the SufaceArea

double Volume(); // Calculates the Volume

void PrintTorus(); // Prints Information about

// the Torus

void BigRadiusFromAreaAndLittleRadius(double, double); // Calculate the BigRadius of

// of a Torus given its Area and LittleRadius

void BigRadiusFromVolumeAndLittleRadius(double, double); // Calculate the BigRadius of

// of a Torus given its Volume and LittleRadius

private:

double big\_radius;

double little\_radius;

double volume;

double surfacearea;

};

Torus::Torus(double bigR, double litR)

{

SetTorus(bigR, litR);

}

void Torus::SetTorus(double bigR, double litR)

{

big\_radius = bigR;

little\_radius = litR;

surfacearea = 4\*PI\*PI\*bigR\*litR;

volume = 2\*PI\*PI\*litR\*litR\*bigR;

}

double Torus::GetBigRadius()

{

return big\_radius;

}

double Torus::GetLittleRadius() {

return little\_radius;

}

double Torus::SurfaceArea()

{

return surfacearea;

}

double Torus::Volume()

{

return volume;

}

void Torus::PrintTorus()

{

cout << "Big Radius: " << big\_radius << endl;

cout << "Little Radius: " << little\_radius << endl;

cout << "Volume: " << volume << endl;

cout << "Surface Area: " << surfacearea << endl;

}

void Torus::BigRadiusFromAreaAndLittleRadius(double area, double litR)

{

double new\_big\_radius = area/(4\*PI\*PI\*litR);

SetTorus(new\_big\_radius, litR);

}

void Torus::BigRadiusFromVolumeAndLittleRadius(double volume, double litR)

{

double new\_big\_radius = volume/(2\*PI\*PI\*litR\*litR);

SetTorus(new\_big\_radius, litR);

}

main()

{

Torus T1(2, 5), T2, T3(5, 10), T4, T5;

T1.PrintTorus();

T2.PrintTorus();

T2.BigRadiusFromVolumeAndLittleRadius(250, 10);

T2.PrintTorus();

T3.PrintTorus();

T4.SetTorus(10, 20);

T4.PrintTorus();

T5.BigRadiusFromAreaAndLittleRadius(500, 5);

T5.PrintTorus();

}

/////////////////////////////////////////////////////////////////////////////

Triangular Pyramid

// Name: Preston Knibbe

//

// Shape: TriPyramid

//

// Equation:

// Surface Area = side^2 \* 2

// Volume = side^2 \* height / 6

#include <cmath>

#include <iostream>

using std::cout;

using std::endl;

using std::fixed;

#include <iomanip>

using std::setw;

using std::setprecision;

using std::showpoint;

using std::setfill;

#define PI 3.14159265358979323846

// Fully implement the following class

class TriPyramid

{

public:

TriPyramid(double = 1); // doubles are the side and height

void SetTriPyramid(double); // doubles are the side and height,

// Should set volume, and area

// too.

double GetSide(); // Return the side

double GetHeight(); // Return the height

double SurfaceArea(); // Calculates the SufaceArea

double Volume(); // Calculates the Volume

void PrintTriPyramid(); // Prints Information about

// the TriPyramid

void SideFromArea(double); // Calculate the side of

// of a TriPyramid given its Area

void SideFromVolume(double); // Calculate the Side of

// of a TriPyramid given its Volume and Height

private:

double side;

double height;

double volume;

double surfacearea;

};

TriPyramid::TriPyramid(double side)

{

SetTriPyramid(side);

}

void TriPyramid::SetTriPyramid(double s)

{

side = s;

height = sqrt((side\*side)-((side\*side)/4));

surfacearea = side \* side \* 2;

volume = (side\*side\*height)/6;

}

double TriPyramid::GetSide()

{

return side;

}

double TriPyramid::GetHeight() {

return height;

}

double TriPyramid::SurfaceArea()

{

return surfacearea;

}

double TriPyramid::Volume()

{

return volume;

}

void TriPyramid::PrintTriPyramid()

{

cout << "Side: " << side << endl;

cout << "Height: " << height << endl;

cout << "Volume: " << volume << endl;

cout << "Surface Area: " << surfacearea << endl;

}

void TriPyramid::SideFromArea(double area)

{

double new\_side = sqrt(area/2);

SetTriPyramid(new\_side);

}

main()

{

TriPyramid T1(100), T2(5), T3, T4;

T1.PrintTriPyramid();

T2.PrintTriPyramid();

T3.SetTriPyramid(15);

T3.PrintTriPyramid();

T4.SideFromArea(60);

T4.PrintTriPyramid();

}

Code Output

Cube 1

HAND CALCULATIONS

V = 2 \* 2 \* 2 = 8

SA = (2)^2 \* 6 = 24

Side: 2

Volume: 8

Surface Area: 24

Side: 1

Volume: 1

Surface Area: 6

Side: 3.46681

Volume: 41.6667

Surface Area: 72.1125

Side: 5

Volume: 125

Surface Area: 150

Side: 10

Volume: 1000

Surface Area: 600

Side: 2.23607

Volume: 11.1803

Surface Area: 30

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

Press any key to continue.

Cube 2

Side: 8

Volume: 512

Surface Area: 384

Side: 1

Volume: 1

Surface Area: 6

Side: 1.65096

Volume: 4.5

Surface Area: 16.3541

Side: 42

Volume: 74088

Surface Area: 10584

Side: 36

Volume: 46656

Surface Area: 7776

Side: 1.77951

Volume: 5.63512

Surface Area: 19

Process returned 0 (0x0) execution time : 0.014 s

Press any key to continue.

Cube 3

Side: 32

Volume: 32768

Surface Area: 6144

Side: 1

Volume: 1

Surface Area: 6

Side: 0.941036

Volume: 0.833333

Surface Area: 5.31329

Side: 60

Volume: 216000

Surface Area: 21600

Side: 12

Volume: 1728

Surface Area: 864

Side: 1.63299

Volume: 4.35465

Surface Area: 16

Process returned 0 (0x0) execution time : 0.019 s

Press any key to continue.

Cylinder 1

HAND CALCULATIONS

SA= 2 \* pi \* 2 \* (2+5)

SA = 28pi

SA = 87.96

V = pi \* 2^2 \* 5

V = 20pi

V = 62.83

Radius: 2

Height: 5

Volume: 62.8319

Surface Area: 87.9646

Radius: 1

Height: 1

Volume: 3.14159

Surface Area: 12.5664

Radius: 10

Height: 0.795775

Volume: 250

Surface Area: 678.319

Radius: 5

Height: 10

Volume: 785.398

Surface Area: 471.239

Radius: 10

Height: 20

Volume: 6283.19

Surface Area: 1884.96

Radius: 5

Height: 10.9155

Volume: 857.301

Surface Area: 500

Process returned 0 (0x0) execution time : 0.012 s

Press any key to continue.

Cylinder 2

Radius: 6

Height: 10

Volume: 1130.97

Surface Area: 603.186

Radius: 1

Height: 1

Volume: 3.14159

Surface Area: 12.5664

Radius: 2

Height: 3.8993

Volume: 49

Surface Area: 74.1327

Radius: 20

Height: 30

Volume: 37699.1

Surface Area: 6283.19

Radius: 5

Height: 26

Volume: 2042.04

Surface Area: 973.894

Radius: 3

Height: 18.2207

Volume: 515.177

Surface Area: 400

Process returned 0 (0x0) execution time : 0.020 s

Press any key to continue.

Cylinder 3

Radius: 8

Height: 12

Volume: 2412.74

Surface Area: 1005.31

Radius: 1

Height: 1

Volume: 3.14159

Surface Area: 12.5664

Radius: 2

Height: 2.22817

Volume: 28

Surface Area: 53.1327

Radius: 25

Height: 39

Volume: 76576.3

Surface Area: 10053.1

Radius: 5

Height: 31

Volume: 2434.73

Surface Area: 1130.97

Radius: 2

Height: 25.8521

Volume: 324.867

Surface Area: 350

Process returned 0 (0x0) execution time : 0.021 s

Press any key to continue.

Torus 1

HAND CALCULATIONS

V = 2 \* pi^2 \* 2 \* 5^2

V = 100pi^2

V = 986.96

SA = 4 \* pi^2 \* 2 \* 5

SA = 40pi^2

SA = 394.78

Big Radius: 2

Little Radius: 5

Volume: 986.96

Surface Area: 394.784

Big Radius: 1

Little Radius: 1

Volume: 19.7392

Surface Area: 39.4784

Big Radius: 0.126651

Little Radius: 10

Volume: 250

Surface Area: 50

Big Radius: 5

Little Radius: 10

Volume: 9869.6

Surface Area: 1973.92

Big Radius: 10

Little Radius: 20

Volume: 78956.8

Surface Area: 7895.68

Big Radius: 2.53303

Little Radius: 5

Volume: 1250

Surface Area: 500

Process returned 0 (0x0) execution time : 0.011 s

Press any key to continue.

Torus 2

Big Radius: 2

Little Radius: 10

Volume: 3947.84

Surface Area: 789.568

Big Radius: 1

Little Radius: 1

Volume: 19.7392

Surface Area: 39.4784

Big Radius: 0.0135095

Little Radius: 15

Volume: 60

Surface Area: 8

Big Radius: 3

Little Radius: 8

Volume: 3789.93

Surface Area: 947.482

Big Radius: 12

Little Radius: 220

Volume: 1.14645e+007

Surface Area: 104223

Big Radius: 0.455945

Little Radius: 3

Volume: 81

Surface Area: 54

Process returned 0 (0x0) execution time : 0.019 s

Press any key to continue.

Torus 3

Big Radius: 5

Little Radius: 97

Volume: 928631

Surface Area: 19147

Big Radius: 1

Little Radius: 1

Volume: 19.7392

Surface Area: 39.4784

Big Radius: 0.000561336

Little Radius: 19

Volume: 4

Surface Area: 0.421053

Big Radius: 6

Little Radius: 9

Volume: 9593.26

Surface Area: 2131.83

Big Radius: 12

Little Radius: 36

Volume: 306984

Surface Area: 17054.7

Big Radius: 0.0220374

Little Radius: 100

Volume: 4350

Surface Area: 87

Process returned 0 (0x0) execution time : 0.021 s

Press any key to continue.

Triangular Pyramid 1

HAND CALCULATIONS

V = (1/3) \* (100^2 / 2) \* (100^2 - (100^2 / 4))^1/2

V = (1/3) \* 5000 \* 86.603

V = 144338

SA = (100^2 / 2) \* 4

SA = 100^2 \* 2

SA = 20000

Side: 100

Height: 86.6025

Volume: 144338

Surface Area: 20000

Side: 5

Height: 4.33013

Volume: 18.0422

Surface Area: 50

Side: 15

Height: 12.9904

Volume: 487.139

Surface Area: 450

Side: 5.47723

Height: 4.74342

Volume: 23.7171

Surface Area: 60

Process returned 0 (0x0) execution time : 0.008 s

Press any key to continue.

Triangular Pyramid 2

Side: 56

Height: 48.4974

Volume: 25348

Surface Area: 6272

Side: 13

Height: 11.2583

Volume: 317.11

Surface Area: 338

Side: 42

Height: 36.3731

Volume: 10693.7

Surface Area: 3528

Side: 8.57321

Height: 7.42462

Volume: 90.9516

Surface Area: 147

Process returned 0 (0x0) execution time : 0.016 s

Press any key to continue.

Triangular Pyramid 3

Side: 43

Height: 37.2391

Volume: 11475.8

Surface Area: 3698

Side: 88

Height: 76.2102

Volume: 98362

Surface Area: 15488

Side: 22

Height: 19.0526

Volume: 1536.91

Surface Area: 968

Side: 16.8077

Height: 14.5559

Volume: 685.342

Surface Area: 565

Process returned 0 (0x0) execution time : 0.017 s

Press any key to continue.