Menu Assignment 2 - Robert P.

Test Outputs:

Source Code:

menu.C (modified) - /home/students/e_prevost/homeworkAssignments/

File Edit Search Preferences Shell Macro Windows

```
/*
Code made by: Robert Prevost

Menu program that outputs the surface area and volume of each specified menu item

Updated on: 10/27/2023
*/

#include <iostream>
#include <cmath>
#include <string>
#include <climits>
using namespace std;

const int CubeN = 1;
const int SphereN = 2;
const int PrismN = 3;
const int CylinderN = 4;
const int ConeN = 5;
const int Quit = 6;

class Cube
{
        private:
                double sidelength;
        public:
                Cube();
                bool SetSL(double sl);
                double GetSL();
                void SurfaceArea();
                void Volume();

};
Cube::Cube()
{
        sidelength = 0.0;
}
bool Cube::SetSL(double sl)
{
        if(sl >= 0.0){
                sidelength = sl;
                return true;
        }
        else{
                return false;
        }
}
```

```
double Cube::GetSL()
{
	return sidelength;
}
void Cube::SurfaceArea()
{
	cout<< "Surface Area of cube: " << pow(sidelength,2.0)*6.0 << endl;
}
void Cube::Volume()
{
	cout << "Volume of cube: " << pow(sidelength,3.0)  << endl;
}

class Sphere
{
	private:
		double r;
	public:
		Sphere();
		bool SetR(double radius);
		double GetR();
		void SurfaceArea();
		void Volume();
};
Sphere::Sphere()
{
	r = 0.0;
}
bool Sphere::SetR(double radius)
{
	if(radius >= 0.0){
		r = radius;
		return true;
	}
	else{
		return false;
	}
}
double Sphere::GetR()
{
	return r;
}
void Sphere::SurfaceArea()
{
	cout << "Surface area of sphere: " << pow(r,2.0)*M_PI*4.0 << endl;
}
void Sphere::Volume()
```

```
{
        cout << "Volume of sphere: " << pow(r,3.0)*M_PI*(4.0/3.0) << endl;
}
class Prism
{
        private:
                double baseArea;
                double basePerim;
                double h;
        public:
                Prism();
                bool SetBaseArea(double ba);
                double GetBaseArea();
                bool SetBasePerim(double bp);
                double GetBasePerim();
                bool SetH(double height);
                double GetH();
                void SurfaceArea();
                void Volume();
};
Prism::Prism()
{
        baseArea = 0.0;
        basePerim = 0.0;
        h = 0.0;
}
bool Prism::SetBaseArea(double ba)
{
        if(ba >= 0.0){
                baseArea = ba;
                return true;
        }
        else{
                return false;
        }
}
bool Prism::SetBasePerim(double bp)
{
        if(bp >= 0.0){
                basePerim = bp;
                return true;
        }
        else{
                return false;
        }
}
bool Prism::SetH(double height)
```

```
{
        if(height >= 0.0){
                h = height;
                return true;
        }
        else{
                return false;
        }
}
double Prism::GetBaseArea()
{
        return baseArea;
}
double Prism::GetBasePerim()
{
        return basePerim;
}
double Prism::GetH()
{
        return h;
}
void Prism::SurfaceArea()
{
        cout << "Surface Area of prism: " << (2.0*baseArea) + (basePerim*h) << endl;
}
void Prism::Volume()
{
        cout << "Volume of prism: "<< baseArea*h << endl;
}
class Cylinder
{
        private:
                double r;
                double h;
        public:
                Cylinder();
                bool SetR(double radius);
                double GetR();
                bool SetH(double height);
                double GetH();
                void SurfaceArea();
                void Volume();
};
Cylinder::Cylinder()
{
        r = 0.0;
        h = 0.0;
```

```
}
bool Cylinder::SetR(double radius)
{
        if(radius >= 0.0){
                r = radius;
                return true;
        }
        else{
                return false;
        }
}
double Cylinder::GetR()
{
        return r;
}
bool Cylinder::SetH(double height)
{
        if(height >= 0.0){
                h = height;
                return true;
        }
        else{
                return false;
        }
}
double Cylinder::GetH()
{
        return h;
}
void Cylinder::SurfaceArea()
{
        cout << "Surface Area of cylinder: " << M_PI*2.0*r*h + 2.0*M_PI*pow(r,2.0) << endl;
}
void Cylinder::Volume()
{
        cout << "Volume of cylinder: " << M_PI*pow(r,2.0)*h << endl;
}
class Cone
{
        private:
                double r;
                double h;
        public:
                Cone();
                bool SetR(double radius);
                double GetR();
                bool SetH(double height);
```

```
                double GetH();
                void SurfaceArea();
                void Volume();
};
Cone::Cone()
{
        r = 0.0;
        h = 0.0;
}
bool Cone::SetR(double radius)
{
        if(radius >= 0.0){
                r = radius;
                return true;
        }
        else{
                return false;
        }
}
double Cone::GetR()
{
        return r;
}
bool Cone::SetH(double height)
{
        if(height >= 0.0){
                h = height;
                return true;
        }
        else{
                return false;
        }
}
double Cone::GetH()
{
        return h;
}
void Cone::SurfaceArea()
{
        double l = r + sqrt(pow(h,2.0) + pow(r,2.0));
        cout << "Surface Area of cone: " << M_PI*r*l << endl;
}
void Cone::Volume()
{
        cout << "Volume of cone: " << (1.0/3.0)*M_PI*pow(r,2.0)*h << endl;
}
```

```
void promptMenu(int choseVal);
int ReadInt (string prompt);
double ReadDouble (string prompt);
void cubeCalc();
void sphereCalc();
void prismCalc();
void cylinderCalc();
void coneCalc();

int main()
{
        int choseVal = 0;
        while(choseVal != Quit){
                cout << "1. Cube\n2. Sphere\n3. Prism\n4. Cylinder\n5. Cone\n6. Quit\n";
                string prompt = "Choose a shape(1-5): ";
                choseVal = ReadInt(prompt);
                promptMenu(choseVal);
        }
}

void promptMenu(int choseVal)
{
        switch(choseVal){
                        case CubeN:
                                {
                                        cubeCalc();
                                }
                                break;

                        case SphereN:
                                {
                                        sphereCalc();
                                }
                                break;

                        case PrismN:
                                {
                                        prismCalc();
                                }
                                break;

                        case CylinderN:
                                {
                                        cylinderCalc();
                                }
                                break;
```

```
                        case ConeN:
                                {
                                        coneCalc();
                                }
                                break;

                        case Quit:
                                {
                                        cout<< "Bye!\n";
                                }
                                break;
                        default:
                                {
                                        cout<<"error!\n";
                                }
                                break;
                }
}
int ReadInt (string prompt)
{
        int rv = 0.0;
        cout << prompt;
        cin >> rv;

        while (cin.fail()){
                cerr << "Error! Cannot read input.\n";
                cin.clear();
                cin.ignore(INT_MAX, '\n');
                cout << prompt;
                cin >> rv;
        }
        return rv;
}
double ReadDouble (string prompt)
{
        double rv = 0.0;
        cout << prompt;
        cin >> rv;

        while (cin.fail() || rv <= 0){
                cerr << "Error! Cannot read input.\n";
                cin.clear();
                cin.ignore(INT_MAX, '\n');
                cout << prompt;
                cin >> rv;
        }
```

```
        return rv;
}
void cubeCalc()
{
        double sl = ReadDouble("Please enter side length of cube: ");
        Cube testCube;
        testCube.SetSL(sl);
        testCube.SurfaceArea();
        testCube.Volume();
}
void sphereCalc()
{
        double r = ReadDouble("Please enter radius of sphere: ");
        Sphere testSphere;
        testSphere.SetR(r);
        testSphere.SurfaceArea();
        testSphere.Volume();
}
void prismCalc()
{
        double baseArea = ReadDouble("Please enter base area of prism: ");
        double basePerim = ReadDouble("Please enter base perimeter of prism: ");
        double height = ReadDouble("Please enter height of prism: ");
        Prism testPrism;
        testPrism.SetBaseArea(baseArea);
        testPrism.SetBasePerim(basePerim);
        testPrism.SetH(height);
        testPrism.SurfaceArea();
        testPrism.Volume();
}
void cylinderCalc()
{
        double r = ReadDouble("Please enter radius of cylinder: ");
        double h = ReadDouble("Please enter height of cylinder: ");
        Cylinder testCylinder;
        testCylinder.SetR(r);
        testCylinder.SetH(h);
        testCylinder.SurfaceArea();
        testCylinder.Volume();
}
void coneCalc()
{
        double r = ReadDouble("Please enter radius of cone: ");
        double h = ReadDouble("Please enter height of cone: ");
        Cone testCone;
        testCone.SetR(r);
        testCone.SetH(h);
        testCone.SurfaceArea();
        testCone.Volume();
}
```