1.

#include"Ball.h"

Ball::Ball(double r)

{

this -> r=r;

}

Ball::Ball(double r,string color)

{

this -> r=r;

setColor(color);

}

void Ball::setR(double r)

{

this -> r=r;

}

double Ball::getR()const

{

return r;

}

double Ball::getArea()

{

return 4\*3.14159\*r\*r;

}

double Ball::getVolume()

{

return 3.14159\*r\*r\*r\*4/3;

}

string Ball::toString()

{

stringstream ss1;

ss1<<"Ball\n"<<fixed<<setprecision(1)<<"radius:"<<getR()<<", "<<"area:"<<getArea()<<", "<<"volume:"<<getVolume()<<", "<<"color:"<<getColor()<<"\n";

return ss1.str();

}

#ifndef B\_H

#define B\_H

#include"GeometricObject.h"

class Ball:public GeometricObject

{

public:

Ball(double r);

Ball(double r,string color);

void setR(double r);

double getR()const;

double getArea();

double getVolume();

virtual string toString();

private:

double r;

};

#endif

#include"Cube.h"

#include"Ball.h"

int main()

{

string s;

double n;

cin>>n>>s;

Ball ball(n,s);

cin>>n>>s;

Cube cube(n,s);

cout<<ball.toString();

cout<<cube.toString();

}

#include"Cube.h"

Cube::Cube(double l)

{

this -> l=l;

}

Cube::Cube(double l,string color)

{

this -> l=l;

setColor(color);

}

void Cube::setL(double l)

{

this -> l=l;

}

double Cube::getL()const

{

return l;

}

double Cube::getArea()

{

return 6\*l\*l;

}

double Cube::getVolume()

{

return l\*l\*l;

}

string Cube::toString()

{

stringstream ss1;

ss1<<"Cube\n"<<fixed<<setprecision(1)<<"length:"<<getL()<<", "<<"area:"<<getArea()<<", "<<"volume:"<<getVolume()<<", "<<"color:"<<getColor()<<"\n";

return ss1.str();

}

#ifndef C\_H

#define C\_H

#include"GeometricObject.h"

class Cube:public GeometricObject

{

public:

Cube(double l);

Cube(double l,string color);

void setL(double l);

double getL()const;

double getArea();

double getVolume();

virtual string toString();

private:

double l;

};

#endif

#include"GeometricObject.h"

GeometricObject::GeometricObject(string color)

{

this -> color=color;

}

string GeometricObject::getColor()const

{

return color;

}

void GeometricObject::setColor(string color)

{

this -> color=color;

}

#ifndef G\_H

#define G\_H

#include<iostream>

#include<string>

#include<sstream>

#include<iomanip>

using namespace std;

class GeometricObject

{

public:

GeometricObject(string color="white");

string getColor()const;

void setColor(string color);

private:

string color;

};

#endif

2.

#include"Ball.h"

Ball::Ball(double r)

{

this -> r=r;

}

Ball::Ball(double r,string c)

{

this -> r=r;

setC(c);

}

void Ball::setR(double r)

{

this -> r=r;

}

double Ball::getR()const

{

return r;

}

double Ball::getArea()

{

return 4\*3.14159\*r\*r;

}

double Ball::getVolume()

{

return 3.14159\*r\*r\*r\*4/3;

}

void Ball::toString()

{

cout<<"Ball"<<endl;

cout<<fixed<<setprecision(1)<<"radius:"<<getR()<<", ";

}

#ifndef B\_H

#define B\_H

#include"GeometricObject.h"

class Ball:public GeometricObject

{

public:

Ball(double r);

Ball(double r,string c);

void setR(double r);

double getR()const;

virtual double getArea();

virtual double getVolume();

virtual void toString();

private:

double r;

};

#endif

#include"Cube.h"

#include"Ball.h"

void display(GeometricObject &g)

{

cout<<fixed<<setprecision(1)<<"area:"<<g.getArea()<<", "<<"volume:"<<g.getVolume()<<", "<<"color:"<<g.getC()<<endl;

}

int main()

{

string s;

double n;

cin>>n>>s;

Ball b(n,s);

cin>>n>>s;

Cube c(n,s);

b.toString();

display(b);

c.toString();

display(c);

}

#include"Cube.h"

Cube::Cube(double l)

{

this -> l=l;

}

Cube::Cube(double l,string c)

{

this -> l=l;

setC(c);

}

void Cube::setL(double l)

{

this -> l=l;

}

double Cube::getL()const

{

return l;

}

double Cube::getArea()

{

return 6\*l\*l;

}

double Cube::getVolume()

{

return l\*l\*l;

}

void Cube::toString()

{

cout<<"Cube"<<endl;

cout<<fixed<<setprecision(1)<<"length:"<<getL()<<", ";

}

#ifndef C\_H

#define C\_H

#include"GeometricObject.h"

class Cube:public GeometricObject

{

public:

Cube(double l);

Cube(double l,string c);

void setL(double l);

double getL()const;

virtual double getArea();

virtual double getVolume();

virtual void toString();

private:

double l;

};

#endif

#include"GeometricObject.h"

GeometricObject::GeometricObject(string c)

{

this -> c=c;

}

string GeometricObject::getC()const

{

return c;

}

void GeometricObject::setC(string c)

{

this -> c=c;

}

void GeometricObject::toString()

{

;

}

#ifndef G\_H

#define G\_H

#include<iostream>

#include<string>

#include<sstream>

#include<iomanip>

using namespace std;

class GeometricObject

{

public:

GeometricObject(string c="white");

string getC()const;

void setC(string c);

virtual double getArea()=0;

virtual double getVolume()=0;

virtual void toString();

private:

string c;

};

#endif