# Код

#include <iostream>

#define PI 3.14159256

class Shape {

public:

double x, y, z; //anchor point

Shape();

Shape(double, double, double);

void printAnchor();

void moveAnchor(double, double, double);

virtual double voulume();

virtual void scale(double);

};

class Sector :public Shape {

public:

double radius;

double angle; //cone angle (width) of sector

double theta, phi; //inclination and azimuth angles of sector axis

Sector();

Sector(double, double, double, double, double, double, double);

void printParameters();

double volume();

void scale(double);

void reflect();

};

Shape::Shape() {

x = 0;

y = 0;

z = 0;

}

Shape::Shape(double x\_, double y\_, double z\_) {

x = x\_;

y = y\_;

z = z\_;

}

void Shape::printAnchor() {

std::cout << "Anchor point is (" << x << ", " << y << ", " << z << ")\n";

}

void Shape::moveAnchor(double x\_, double y\_, double z\_) {

x = x\_;

y = y\_;

z = z\_;

}

double Shape::voulume() {

return 0;

}

void Shape::scale(double s) {

}

Sector::Sector() :Shape() {

radius = 0;

angle = 0;

theta = 0;

phi = 0;

}

Sector::Sector(double x\_, double y\_, double z\_, double r\_, double angle\_, double theta\_ = 0, double phi\_ = 0) :Shape(x\_, y\_, z\_) {

radius = r\_;

angle = angle\_;

theta = theta\_;

phi = phi\_;

}

void Sector::printParameters() {

printAnchor();

std::cout << "Radius is " << radius << "\n";

std::cout << "Cone angle is " << angle << "\n";

std::cout << "Sector axis inclination and azimuth angles are " << theta << ", " << phi << "\n\n";

}

double Sector::volume() {

return (2. / 3.) \* PI \* pow(radius, 3) \* (1 - cos(angle));

}

void Sector::scale(double s) {

if (s < 1) {

radius \*= s;

}

else {

std::cerr << "Scale factor should be <1\n\n";

}

}

void Sector::reflect() {

theta = PI - theta;

if (phi < PI) {

phi += PI;

}

else {

phi -= PI;

}

}

int main()

{

Sector a{ 0.,0.,1.,10.,0.1, 0.66 };

a.printParameters();

a.moveAnchor(2, 2, 2);

a.reflect();

a.scale(1.3);

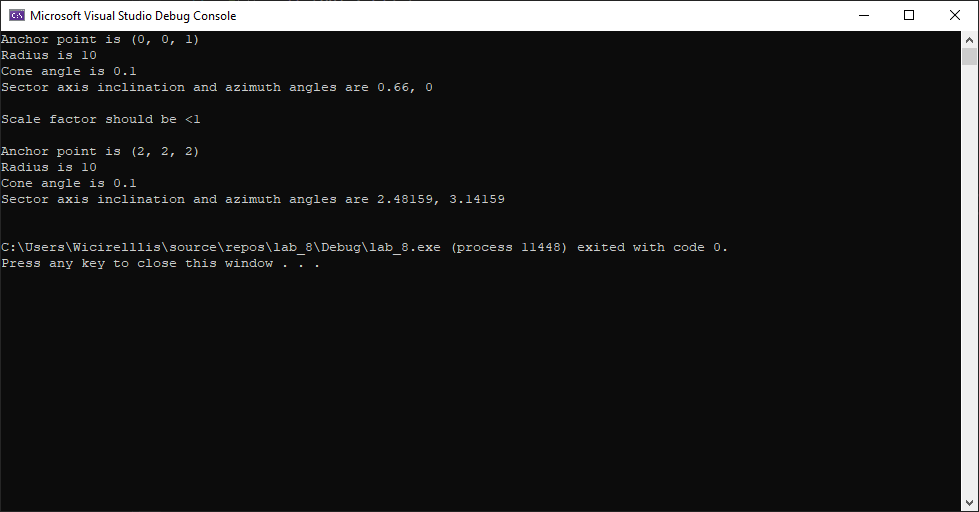
a.printParameters();

return 0;

}

# 2.

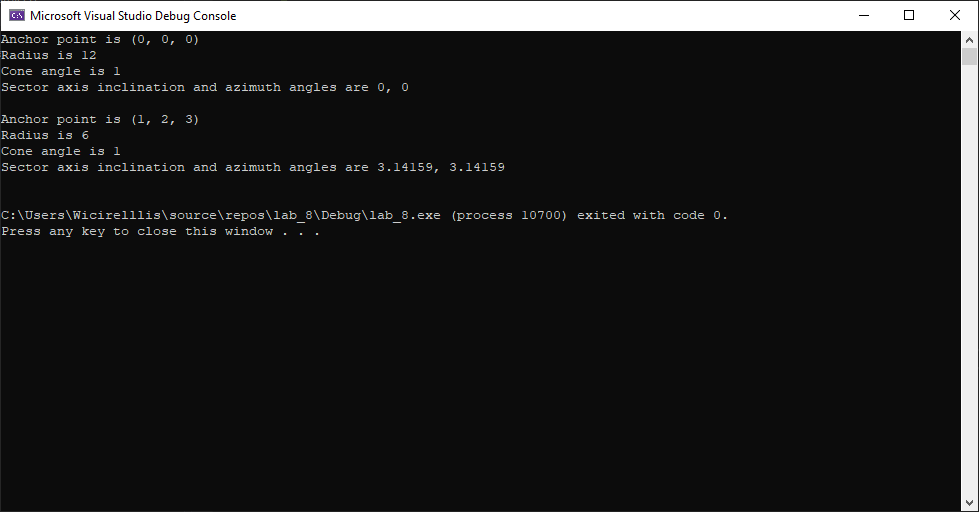
Проверим работу программы. Кусок в мейне служит ровно этому.



Работает (по крайней мере на таких данных) верно.

# 3.

Еще один скрин с другими числами.



# 4. Вопросы

1) Что такое виртуальная функция? Зачем она нужна?  
Виртуальные функции можно переопределять в классах-наследниках.

2) К каким данным базового класса имеет доступ порожденный класс?

К public и protected полям\методам. Доступа к private нет.