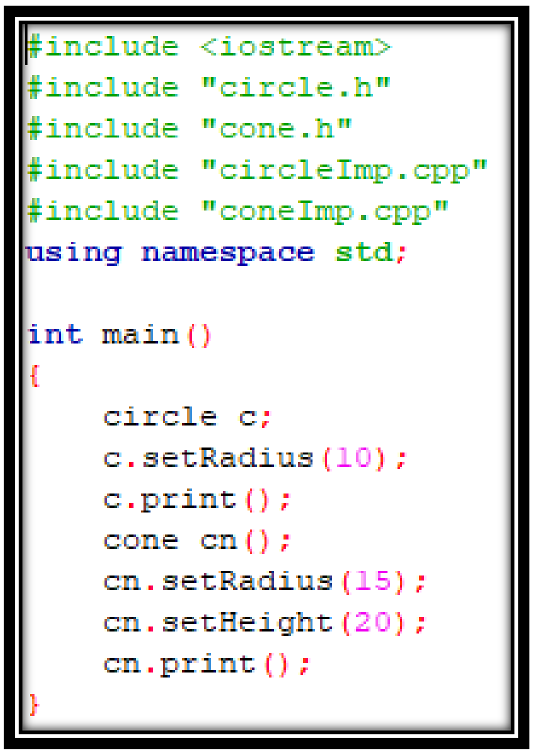
# Lab Exercise # 10

**Name:** Josiah Joed G.  Getes

**Course & Year:** BSCS – 1

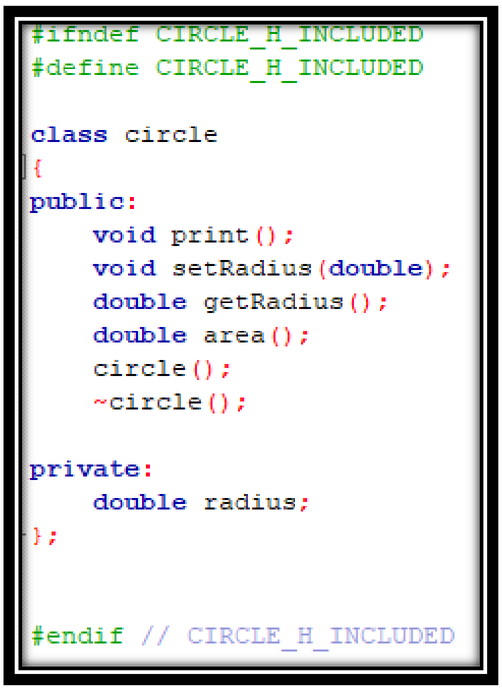
**Objectives:** To create a base class for a circle shape with the appropriate member variables and functions.
To derive another class from the base class for a related shape, then override some function from it.
To implement the function of the base class and the derived class

**CODE:**

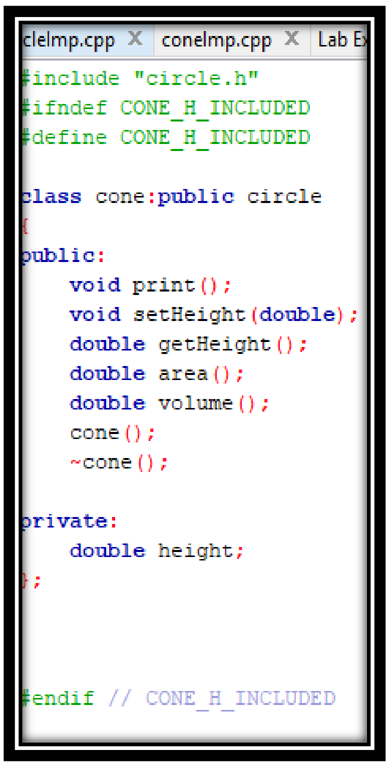


```
#include <iostream>
#include "circle.h"
#include "cone.h"
#include "circleImp.cpp"
#include "coneImp.cpp"
using namespace std;

int main()
{
    circle c;
    c.setRadius(10);
    c.print();
    cone cn();
    cn.setRadius(15);
    cn.setHeight(20);
    cn.print();
}
```

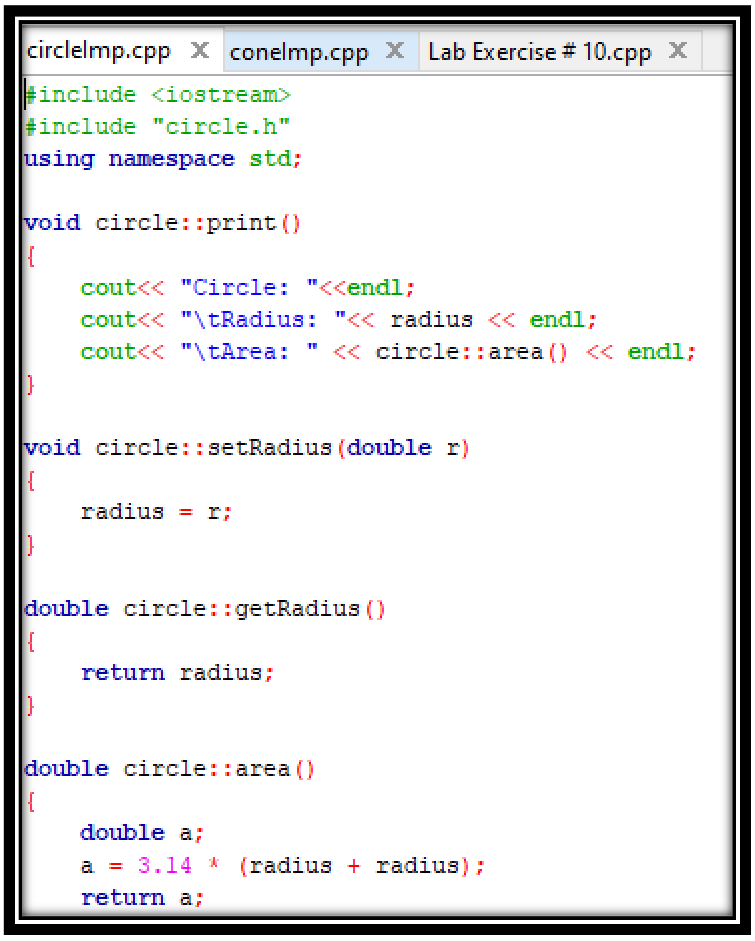
**HEADER FILES:**



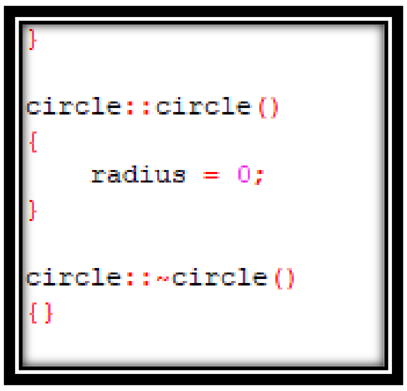
```
#ifndef CIRCLE_H_INCLUDED
#define CIRCLE_H_INCLUDED

class circle
{
public:
    void print();
    void setRadius(double);
    double getRadius();
    double area();
    circle();
    ~circle();

private:
    double radius;
};


#endif // CIRCLE_H_INCLUDED
```



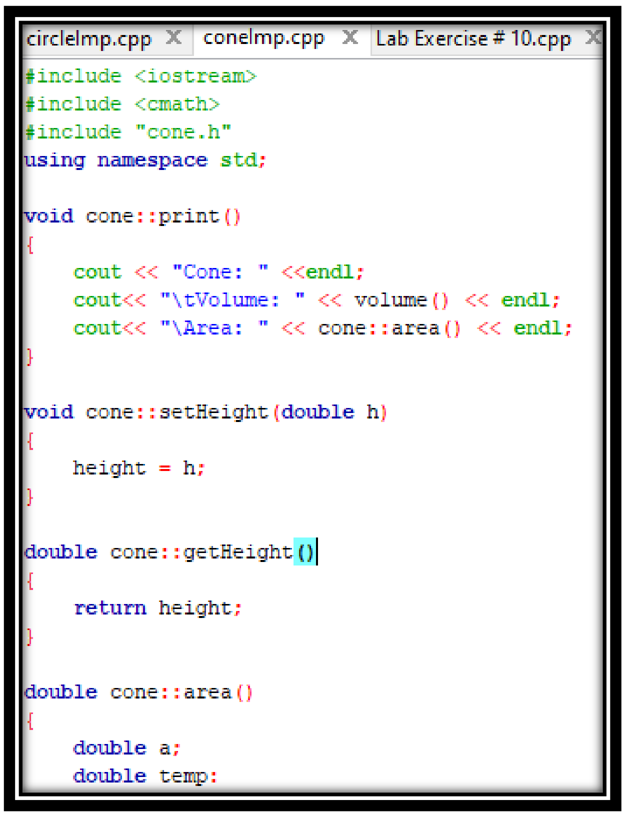
```
#include "circle.h"
#ifndef CONE_H_INCLUDED
#define CONE_H_INCLUDED

class cone:public circle
{
public:
    void print();
    void setHeight(double);
    double getHeight();
    double area();
    double volume();
    cone();
    ~cone();

private:
    double height;
};


#endif // CONE_H_INCLUDED
```



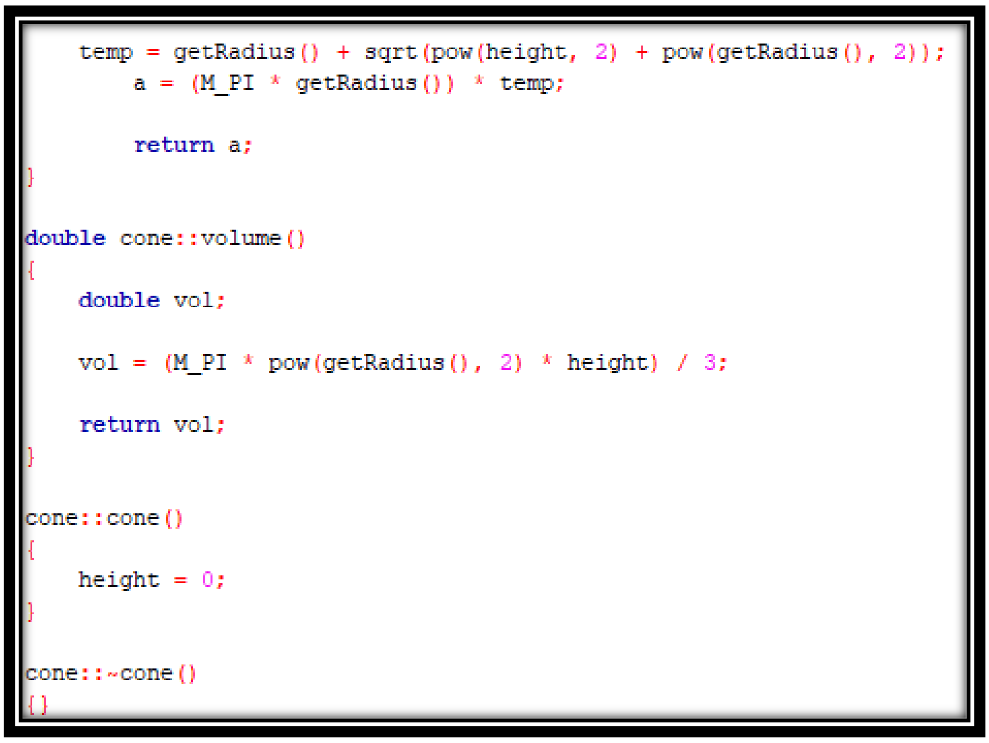
```
#include <iostream>
#include "circle.h"
using namespace std;

void circle::print()
{
    cout<< "Circle: "<<endl;
    cout<< "\tRadius: "<< radius << endl;
    cout<< "\tArea: " << circle::area() << endl;
}

void circle::setRadius(double r)
{
    radius = r;
}

double circle::getRadius()
{
    return radius;
}

double circle::area()
{
    double a;
    a = 3.14 * (radius + radius);
    return a;
```



```
}

circle::circle()
{
    radius = 0;
}

circle::~circle()
{}
```

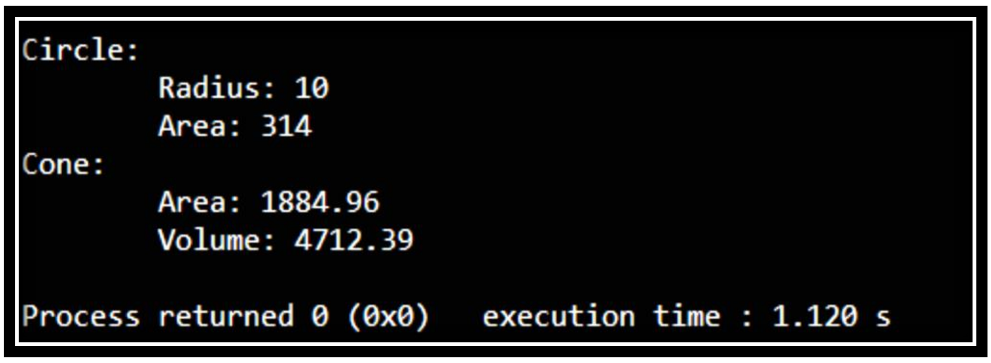


```
#include <iostream>
#include <cmath>
#include "cone.h"
using namespace std;

void cone::print()
{
    cout << "Cone: " <<endl;
    cout<< "\tVolume: " << volume() << endl;
    cout<< "\Area: " << cone::area() << endl;
}

void cone::setHeight(double h)
{
    height = h;
}

double cone::getHeight()
{
    return height;
}

double cone::area()
{
    double a;
    double temp;
```



```
    temp = getRadius() + sqrt(pow(height, 2) + pow(getRadius(), 2));
        a = (M_PI * getRadius()) * temp;

        return a;
}

double cone::volume()
{
    double vol;

    vol = (M_PI * pow(getRadius(), 2) * height) / 3;

    return vol;
}

cone::cone()
{
    height = 0;
}

cone::~cone()
{}
```

**OUTPUT:**



```
Circle:
        Radius: 10
        Area: 314
Cone:
        Area: 1884.96
        Volume: 4712.39

Process returned 0 (0x0)   execution time : 1.120 s
```