**Assignment 1:**

#include <iostream>

#include <cmath>

using namespace std;

class Shape {

protected:

    string name;

public:

    Shape(string n) : name(n) {}

    virtual void calculateArea() = 0;

    virtual ~Shape() {

        cout << "Destroying Shape: " << name << endl;

    }

};

class Rectangle : public Shape {

    float length, breadth;

public:

    Rectangle(float l, float b) : Shape("Rectangle"), length(l), breadth(b) {}

    void calculateArea() override {

        float area = length \* breadth;

        cout << name << " Area = " << area << endl;

    }

    ~Rectangle() {

        cout << "Destroying Rectangle" << endl;

    }

};

class Circle : public Shape {

    float radius;

public:

    Circle(float r) : Shape("Circle"), radius(r) {}

    void calculateArea() override {

        float area = M\_PI \* radius \* radius;

        cout << name << " Area = " << area << endl;

    }

    ~Circle() {

        cout << "Destroying Circle" << endl;

    }

};

class Triangle : public Shape {

    float base, height;

public:

    Triangle(float b, float h) : Shape("Triangle"), base(b), height(h) {}

    void calculateArea() override {

        float area = 0.5 \* base \* height;

        cout << name << " Area = " << area << endl;

    }

    ~Triangle() {

        cout << "Destroying Triangle" << endl;

    }

};

int main() {

    Shape\* shapes[3];

    shapes[0] = new Rectangle(10, 5);

    shapes[1] = new Circle(7);

    shapes[2] = new Triangle(6, 4);

    for (int i = 0; i < 3; ++i) {

        shapes[i]->calculateArea();

    }

    for (int i = 0; i < 3; ++i) {

        delete shapes[i];

    }

    return 0;

}