Week 10-11

Q9. Simple program with abstract class eg class figure, circle, rectangle

#include <iostream>

#include <cmath>

using namespace std;

class Figure { // Abstract base class with pure virtual functions.

public:

virtual double area() = 0; // Pure virtual function for area calculation.

virtual void draw() = 0; // Pure virtual function for drawing shape.

};

class Circle : public Figure { // Circle inherits Figure.

private:

double radius;

public:

Circle(double r) : radius(r) {}

double area() override { // Overriding area calculation for Circle.

return M\_PI \* radius \* radius;

}

void draw() override { // Overriding draw function for Circle.

cout << "Drawing Circle with radius: " << radius << endl;

}

};

class Rectangle : public Figure { // Rectangle inherits Figure.

private:

double width, height;

public:

Rectangle(double w, double h) : width(w), height(h) {}

double area() override { // Overriding area calculation for Rectangle.

return width \* height;

}

void draw() override { // Overriding draw function for Rectangle.

cout << "Drawing Rectangle with width: " << width <<

", height: " << height << endl;

}

};

int main() {

Figure\* fig1 = new Circle(5); // Creating Circle instance.

Figure\* fig2 = new Rectangle(4, 6); // Creating Rectangle instance.

fig1->draw();

cout << "Area: " << fig1->area() << endl;

fig2->draw();

cout << "Area: " << fig2->area() << endl;

delete fig1; // Free memory allocated for Circle.

delete fig2; // Free memory allocated for Rectangle.

return 0;

}