Friend Class for Stream Insertion:

Scenario: You have a Point class with private members for x and y coordinates. You want to define a way to easily print Point objects to output streams like cout.

Create a Point class with private x and y members and a public constructor.

Design a friend class PointOutputStream that has an overloaded << operator to format and insert Point objects into output streams.

In main, demonstrate creating Point objects and printing them using cout.

CODE:

#include <iostream>

using namespace std;

class Point; // Forward declaration

// Friend class declaration

class PointOutputStream {

public:

// Overloaded << operator for output

friend ostream& operator<<(ostream& os, const Point& point);

};

// Declaration of the Point class

class Point {

private:

int x, y; // Private members for x and y coordinates

public:

// Constructor to initialize x and y

Point(int xCoord, int yCoord) : x(xCoord), y(yCoord) {}

// Friend class declaration

friend class PointOutputStream;

};

// Definition of the overloaded << operator

ostream& operator<<(ostream& os, const Point& point) {

// Output the point in the format (x, y)

os << "(" << point.x << ", " << point.y << ")";

return os;

}

int main() {

// Create Point objects

Point p1(10, 20);

Point p2(30, 40);

// Use the overloaded << operator to print Point objects

cout << "Point 1: " << p1 << endl;

cout << "Point 2: " << p2 << endl;

return 0;

}