/\*\*\*\*\*\*\*\*\*\* Online on Operator Overloading \*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Time: 30 minutes \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Add the required functionalities so all the operations in main() works \*/

#include<iostream>

#include<cmath>

#include<string>

using namespace std;

void printVar(string s)

{

std::cout << s << " = ";

}

class Point2D;

class Vector2D;

/\* The Point2D class \*/

class Point2D

{

double x,y;

public:

Point2D()

{

x = 0;

y = 0;

}

Point2D(double, double);

void setX(double);

void setY(double);

double getX();

double getY();

void print();

~Point2D();

// your code

};

Point2D::Point2D(double argx,double argy)

{

x = argx;

y = argy;

}

void Point2D::setX(double argx)

{

x = argx;

}

void Point2D::setY(double argy)

{

y = argy;

}

double Point2D::getX()

{

return x;

}

double Point2D::getY()

{

return y;

}

void Point2D::print()

{

std::cout << "(" << x << "," << y << ")\n";

}

Point2D::~Point2D()

{

x = 0;

y = 0;

}

/\* The Vector2D class \*/

class Vector2D

{

double x, y;

public:

Vector2D(double x=0, double y=0) { this->x = x; this->y = y; }

/\* A utility function to print a 2D Vector in the format xi + yj \*/

/\* You need not modify this function \*/

void print() {

if (!(this->x || this->y)) { std::cout<<0<<endl; return; }

if(this->x) std::cout<<this->x<<"i";

if(this->x && this->y) std::cout<<(this->y>0?" + ":" - ");

if(this->y) std::cout<<(this->x?abs(this->y):this->y)<<"j";

std::cout << endl;

}

// your code

};

// your code

int main()

{

Point2D p1(5,10), p2(15,10);

printVar("Initial p1"); p1.print(); // Initial p1 = (5,10)

printVar("Initial p2"); p2.print(); // Initial p2 = (15,10)

Vector2D v1(3,-4), v2(0,5);

printVar("Initial v1"); v1.print(); // Initial v1 = 3i - 4j

printVar("Initial v2"); v2.print(); // Initial v2 = 5j

p1++; // should increment the value of y coordinate by 1

printVar("p1++"); p1.print(); // p1++ = (5,11)

++p1; // should increment the value of x coordinate by 1

printVar("++p1"); p1.print(); // ++p1 = (6,11)

Point2D p3 = p2 + v1; // should add the respective components of the point and vector, e.g. p2(15,10) + v1(3,-4) = p3(15+3, 10-4)

printVar("Initial p3"); p3.print(); // Initial p3 = (18,6)

Point2D p4 = p3 + v2; // should add the respective components of the point and vector

printVar("Initial p4"); p4.print(); // Initial p4 = (18,11)

return 0;

}