

Point.h

```
#ifndef POINT_H  
#define POINT_H
```

```
class Point {
```

```
public:
```

```
    Point();
```

```
    Point(double xCoordinate, double yCoordinate);
```

```
    double getXCoord();
```

```
    double getYCoord();
```

```
    Point& operator++();
```

```
    Point& operator++(int dummy);
```

```
private:
```

```
    double xCoordinate;
```

```
    double yCoordinate;
```

```
};
```

```
#endif
```

Point.cpp

```
#include "Point.h"
```

```
using namespace std;
```

```
Point::Point(double xCoord, double yCoord)  
: xCoordinate(xCoord), yCoordinate(yCoord)
```

```
{ }
```

```
Point::Point(): xCoordinate(0.0), yCoordinate(0.0)
```

```
{ }
```

```
double Point::getXCoord() { return xCoordinate; }
```

```
double Point::getYCoord() { return yCoordinate; }
```

```
Point& Point::operator++() {
```

```
    xCoordinate++;
```

```
    yCoordinate++;
```

```
    return *this;
```

The test program should do the following:

1. Create the point *p1* to be (0,0).
2. Create the point *p2* to be (2, 4).
3. Print the x and y coordinates for both.
4. Use ++ to update the point (0,0) to be the point (1,1). (Make sure that you call the correct version of ++!)
5. Print the new values of the coordinates of *p1*.

```
#include <iostream>
```

```
#include Point "Point.h"
```

```
using namespace std;
```

```
int main() {
```

```
    Point p1;
```

```
    Point p2(2.0, 4.0);
```

```
    cout << "p1 is (" << p1.getXCoord()
         << " , " << p1.getYCoord() << " )"
         << endl;
```

```
    //same for p2
```

```
    ++p1;
```

```
    cout << again
```