C++ code kata: Week #6

Hello 👋

In two-dimensional **Euclidean** space, a **point** is represented by an ordered pair (x, y) of numbers, where the first number conventionally represents the horizontal and is often denoted by x, and the second number conventionally represents the vertical and is often denoted by y. .

Ground Exercise

In today's exercise 🏃 you have to create a class called `Point` that may take **zero or two integers** as arguments. The points can be initialized while the creation or later. You also have to write a method such that point can identify its quadrant. The quadrants will go as follows

X-coordinate	Y-coordinate	Quadrant
Positive	Positive	First
Negative	Positive	Second
Negative	Negative	Third
Positive	Negative	Fourth

The example below shows how we write `Point()`. e.g. 👇

```
auto pt1 = Point(12, 4);

cout << pt1;
// x:12 y:4
cout << pt1.quadrant()
// First
```

Moreover, a new `Point` can be created from an existing point. The point class should also have the ability to provide console output.

```
auto pt2 = Point(-34, 56);
auto pt3 = Point(pt2);

cout << pt2;
// x:-34 y:56
```

```
cout << pt3;
// x:-34 y:56

pt3.x(45);

cout << pt2;
// x:-34 y:56

cout << pt3;
// x:45 y:56

cout << pt2.quadrant();
// Second

cout << pt3.quadrant();
// First
```

Rules 📜

1. You have to pass either zero or two integers only while object creation.
2. All co-ordinates are in integers only.
3. The return value will be either true or False.
4. You cannot use any library other than the C/ C++ standard library.

Tools 🛠

To write solutions you can download tools from:

Tool	Usage	Download Location
Visual Studio Code	Lightweight program editor.	Download
GCC (Linux)	C / C++ Compiler.	Download
MingGW (Windows)	C / C++ Compiler.	Download
XCode toolkit (MacOS)	IDE with multi language support.	Download
Catch2	C++ Test framework	Download

Test Script 💉

Test Script for this week's problem can be downloaded from **here**.

Sensei Says 🥋

"What you learn is not what you read or listened to, but rather what you attempted at..."

Progressive learning

If you feel the exercise a little bit difficult to solve do not get disheartened. The whole idea behind these programming exercises is not to solve them but rather attempt them.

Try to attempt them in as many ways as possible, you will learn new techniques that will be very helpful to you in the long run especially in the work field.

We will present you with **solution mail/document** also. The solution will show you various ways to solve a problem and why a technique is better than the last one.

We encourage you to make notes from the solution provided and try to apply what you have learnt in the future exercise.

What I will gain from these exercises ?

1. A better and faster way to solve the exercise.
2. Reusable components like containers, algorithms etc. that you can apply to the problem at hand rather than designing your own.
3. Confidence and attitude to solve a problem in new ways, instead of trying monotonous techniques.
4. Writing the robust and quality software using **test driven development**.

All the best 🤞

In case you have any difficulty using the tools or some questions related to this exercise. Reach us at

1. 💬 programmingdays (Skype)