

**E30:** From the detailed table below, I took the best times from the methods with a clear bias towards one type (Cartesian data for getX, getY and convertToPolar and Polar data for getRho, getTheta and convertToCartesian), then for the rest of the results I used the median value.

Although they were close, PointCP1 seemed to be quicker, which is not surprising when it comes to switching the storage types, but is more surprising when it comes to the 'get' methods.

		Average Time Per Execution	
		Data Type:	
Method Test:		PointCP1	PointCP5
	getX	3.58E-06	4.17E-06
	getY	3.62E-06	5.85E-06
	getRho	2.22E-06	4.22E-06
	getTheta	3.85E-06	5.83E-06
	toString	1.13E-04	1.14E-04
	convertToPolar	8.12E-04	5.83E-05
	convertToCartesian	6.83E-04	4.23E-05
	rotatePoint	7.16E-05	6.46E-05
	getDistance	4.08E-05	3.13E-05

To get these, 12 tests were ran, each time running each method several thousand times and then averaged them all.

**Detailed Table:** Includes different columns for different types of coordinates since they can lead to different times and allows for more control when deciding how often one may need to, for example, call `getX()` on a polar storage type.

		Average Time Per Execution			
		Data Type:			
		PointCP3	PointCP1, Polar	PointCP1, Cartesian	PointCP2
Method Test:	getX	4.17E-06	2.29E-05	3.58E-06	1.88E-05
	getY	5.85E-06	2.85E-05	3.62E-06	1.96E-05
	getRho	4.61E-06	2.22E-06	2.82E-06	4.22E-06
	getTheta	7.02E-05	3.85E-06	7.67E-05	5.83E-06
	toString	1.10E-04	1.08E-04	1.19E-04	1.18E-04
	convertToPolar	5.83E-05	n/a	8.12E-04	n/a

	<b>convertToCartesian</b>	n/a	6.83E-04	n/a	4.23E-05
	<b>rotatePoint</b>	5.29E-05	7.83E-05	6.48E-05	7.62E-05
	<b>getDistance</b>	5.37E-06	7.95E-05	2.09E-06	5.73E-05