Overriding equals method in Java

All classes in Java inherit from the Object class, directly or indirectly.
The Object class has some basic methods like clone(), toString(), equals(),.. etc. We can override the equals method in our class to check whether two objects have same data or not

```
class Complex {

	private double re, im;

	public Complex(double re, double im) {

		this.re = re;

		this.im = im;

	}

			// Overriding equals() to compare two Complex objects

	@Override

	public boolean equals(Object o) {

		// If the object is compared with itself then return true

		if (o == this) {

			return true;

		}

		/* Check if o is an instance of Complex or not

		"null instanceof [type]" also returns false */

		if  (! (o instanceof Complex)) {

			return false;

		}

		// typecast o to Complex so that we can compare data members

		Complex c = (Complex) o;
```

```
                    // Compare the data members and return accordingly
        return Double.compare(re, c.re) == 0  &&  Double.compare(im, c.im) == 0;
        }
}
```

```
  public class Main {
        public static void main(String[] args) {
                Complex c1 = new Complex(10, 15);
                Complex c2 = new Complex(10, 15);
                if (c1.equals(c2)) {
                        System.out.println("Equal ");
                } else {
                        System.out.println("Not Equal ");
                }
        }
}
```

As a side note, when we override equals(), it is recommended to also override the hashCode() method. If we don't do so, equal objects may get different hash-values; and hash based collections, including HashMap, HashSet, and Hashtable do not work properly