**On Shallow clone and Deep clone**

In object-oriented programming, object copying is creating a copy of an existing object, the resulting object is called an object copy or simply copy of the original object.There are several ways to copy an object, most commonly by a copy constructor or cloning.

We can define Cloning as “create a copy of object”. Shallow, deep and lazy copy is related to cloning process.

These are actually three ways for creating copy object.

**Shallow clone**

* Whenever we use default implementation of clone method we get shallow copy of object means it creates new instance and copies all the field of object to that new instance and returns it as object type, we need to explicitly cast it back to our original object. This is shallow copy of the object.
* clone() method of the object class support shallow copy of the object. If the object contains primitive as well as non primitive or reference type variable in shallow copy, the cloned object also refers to the same object to which the original object refers as only the object references gets copied and not the referred objects themselves.
* That’s why the name shallow copy or shallow cloning in Java. If only primitive type fields or Immutable objects are there then there is no difference between shallow and deep copy in Java.

**Deep Copy or clone**

* Whenever we need own copy not to use default implementation we call it as deep copy, whenever we need deep copy of the object we need to implement according to our need.
* So for deep copy we need to ensure all the member class also implement the Cloneable interface and override the clone() method of the object class.
* A deep copy means actually creating a new array and copying over the values.