How are arrays copied in Java?

Question

I know that double behaves differently than Double but I don't yet know all of the details. For example for:

```
protected double[] _refPoint = new double[3];

public void setReferencePoint(double[] point){
   _refPoint = point;
}
```

Does this do a shallow copy or a deep copy?

Answer

No copy at all, it just changes the `_refPoint` variable to reference the passed in array, and presumably eventually garbage collects the old `double[3]`. This is not in general a good idea, because the code may not behave the way the caller expects, for example:

```
double d = { 1, 2, 3}
setReferencePoint(d);
d[0] = 3;
```

changes the array which is now referenced internally by the class containing `setReferencePoint`. It is probably better practice to internally clone the array, for example:

```
public void setReferencePoint(double[] point){
   _refPoint = point.clone();
}
```

or to make sure the documentation to `setReferencePoint` explains that it will hold a reference to the array and the user should not modify it.

In fact we generally discourage the use of `double[3]` for momentum or space point. Try using Hep3Vector instead:

http://java.freehep.org/freehep-physics/apidocs/hep/physics/vec/Hep3Vector.html