Implementing hashCode :

1. If a class overrides equals, it must override hashCode
2. When they are both overridden, equals and hashCode must use the same set of fields
3. If two objects are equal, then their hashCode values must be equal as well
4. Whenever it is invoked, the hashCode method must return the same integer value.

It is a popular misconception that hashCode provides a unique identifier for an object. It does not.

```
public static void main(String[] args) {
   Integer ii=new Integer(3);
   Boolean bb=true;
   Double dd=new Double(2.2);
   String ss="Fadel";
   String sss="FadelK";
   System.out.println(ii.hashCode());
   System.out.println(bb.hashCode());
   System.out.println(dd.hashCode());
   System.out.println(ss.hashCode());
   System.out.println(sss.hashCode());
}
```

```
3
1231
-644349949
67635536
2096701691
```

```
public class Pen {
  String Color;
  double Volume;
  public String name(){return "I'm a Pen.";}

  public boolean equals(Object o){
    if (o == null) return false;
    if (o == this) return true;
    if (!(o instanceof Pen)) return false;
    Pen OO= (Pen) o;
    if(this.Color.equals(OO.Color) && this.Volume==OO.Volume){return true;}
    else{return false;}
  }

  public int hashCode(){
    Double d=new Double(Volume);
    return d.hashCode() + Color.hashCode();
  }
}
```