# Why we need Override Equals

Let take example: Person { int id; int name; }

```
public static void main( args[] )
{
    Person p1 = new Person( 10, "jyoti");
    Person p2 = new Person( 10 , "jyoti");
    S.O.P ( P1 == P2);  // False
    S.O.P ( P1.equals(P2)); // {False ? }
}
```

As equals method present in the obj class only check reference of objects. That's accoding to it, equals method return false.

We Need to override the Equals Method.

```
public boolean equals( Person P)
{
      if (P== this)
      return false;
   else if ( this.id == P.id &&
            this.name == P.name)
    return true;
}
```

When we override equals method than

S.O.P ( P1.equals(P2) ; ⇒ TRUE

Now our class working fine.

# Why we override the hashcode()

```
LinkedHashSet<Person> ll = new LinkedHashSet<Person>();
  ll.add (P1); ll.add(P2); //
S.O.P ( ll) ; [ 10 jyoti , 10 jyoti]
```

? how Set allow duplicate values.

because hashcode P1 & P2 are different because of that Set treats them different obj although having values are same

override the hashcode().

```
public int hashcode()
 {
      return this.name %10;
 }
```

Now, hascode of P1 & P2 are same Set treat them same object.

```
S.O.P (ll);    [10, jyoti);
```

contract

If two object are equal by equals() then hashcode of two object must be same.

If hashcode of 2 objects are same, still 2obj can be equal or not, depend on equal method.