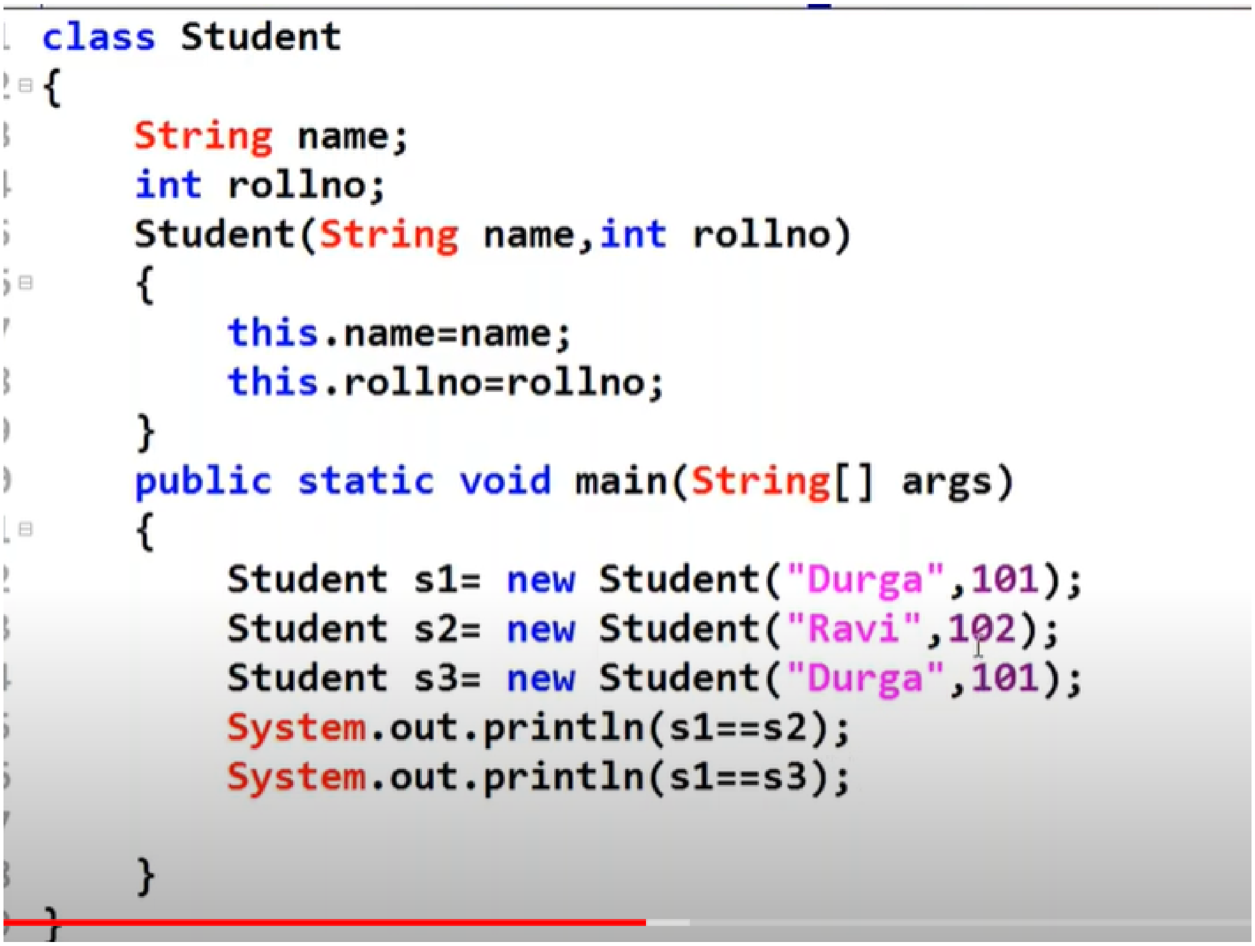
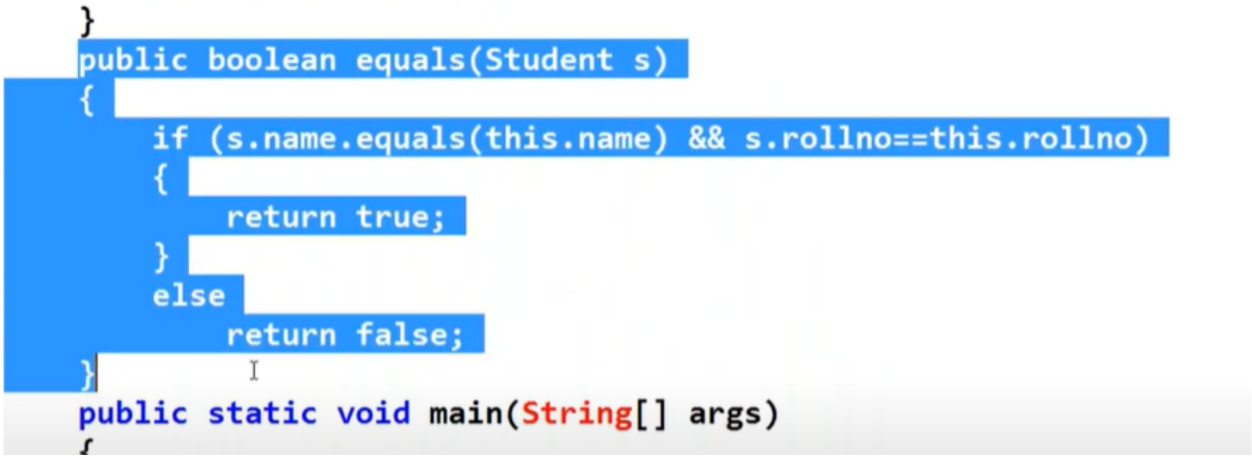


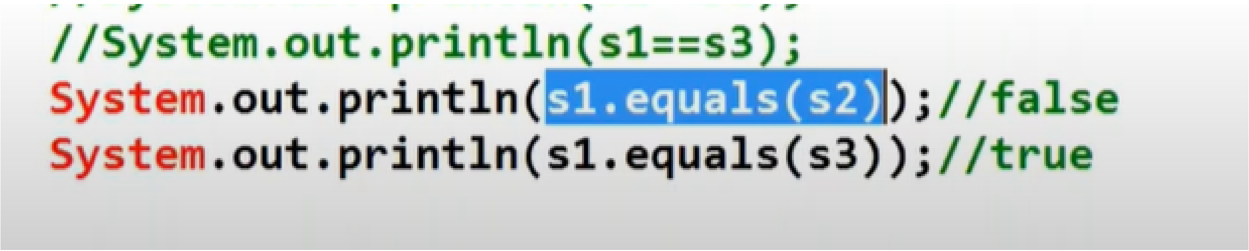
```
class Student
{
    String name;
    int rollno;
    Student(String name,int rollno)
    {
        this.name=name;
        this.rollno=rollno;
    }
    public static void main(String[] args)
    {
        Student s1= new Student("Durga",101);
        Student s2= new Student("Ravi",102);
        Student s3= new Student("Durga",101);
        System.out.println(s1==s2);
        System.out.println(s1==s3);

    }
}
```

ans=> false false



```
    }
    public boolean equals(Student s)
    {
        if (s.name.equals(this.name) && s.rollno==this.rollno)
        {
            return true;
        }
        else
            return false;
    }
    public static void main(String[] args)
    {
```

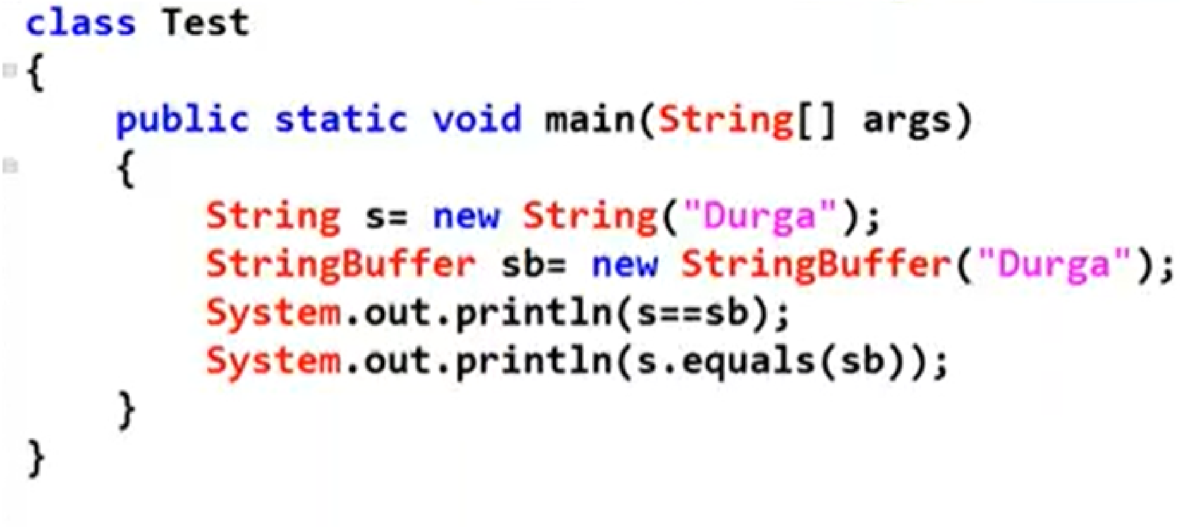
Overridden equals method



```
//System.out.println(s1==s3);
System.out.println(s1.equals(s2));//false
System.out.println(s1.equals(s3));//true
```

now it will give false true

Q1



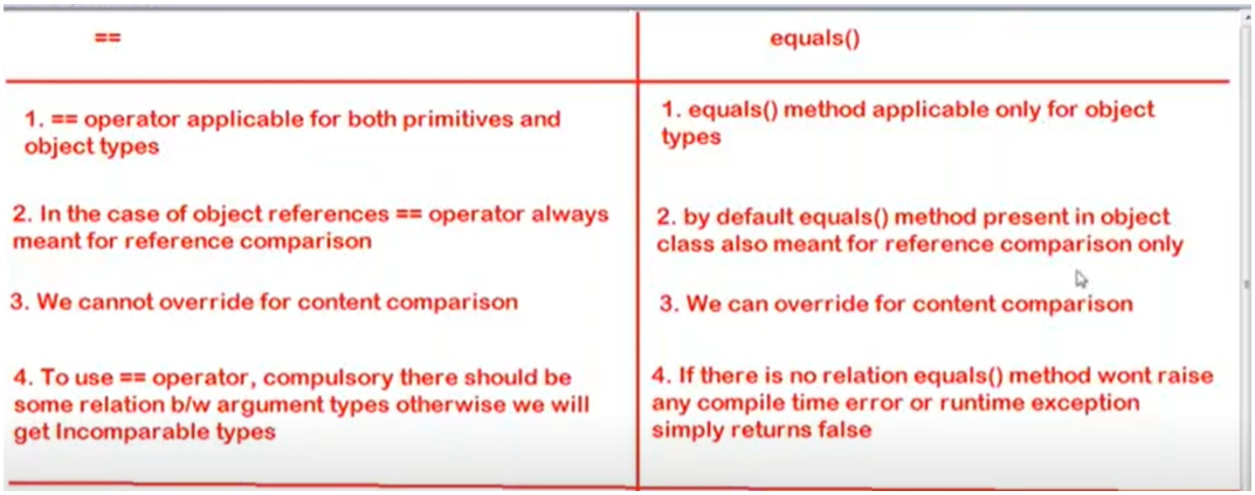
```
class Test
{
    public static void main(String[] args)
    {
        String s= new String("Durga");
        StringBuffer sb= new StringBuffer("Durga");
        System.out.println(s==sb);
        System.out.println(s.equals(sb));
    }
}
```

Compile time error in == because there should be a relationship between those two object parent child or same then only == operation valid.

Comment out == line

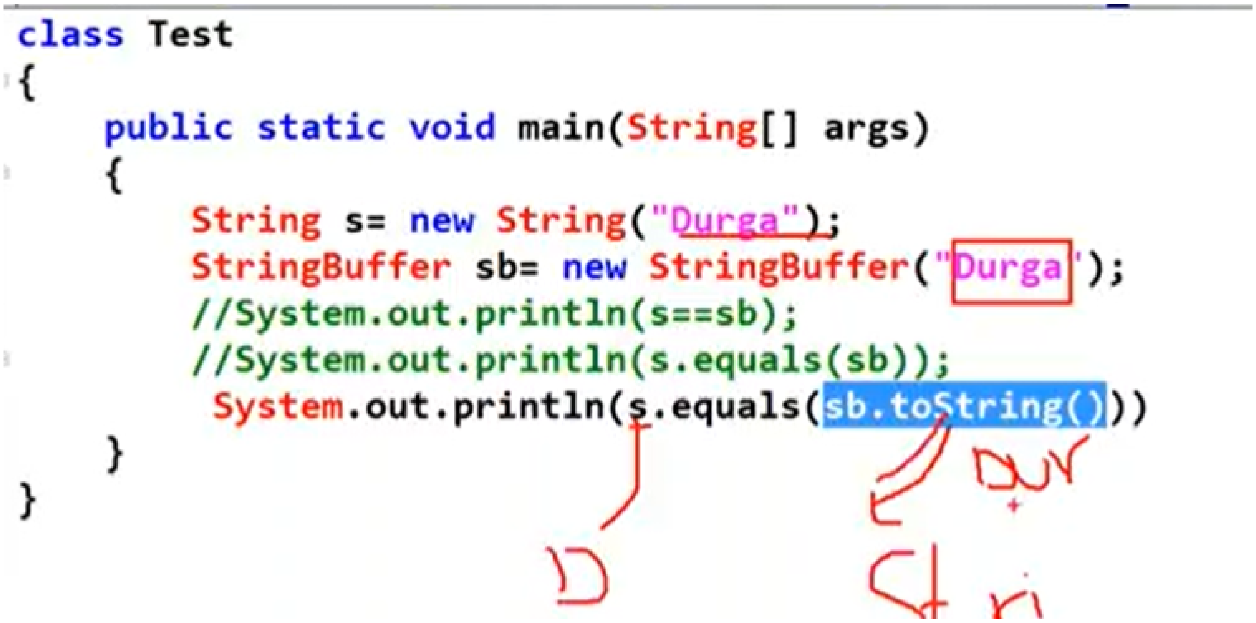
Then it will give false as result because if there is no relation between the two classes, the .equals method will just give the result as false without any comparison.

Different between == and equals()



==	equals()
1. == operator applicable for both primitives and object types	1. equals() method applicable only for object types
2. In the case of object references == operator always meant for reference comparison	2. by default equals() method present in object class also meant for reference comparison only
3. We cannot override for content comparison	3. We can override for content comparison
4. To use == operator, compulsory there should be some relation b/w argument types otherwise we will get Incomparable types	4. If there is no relation equals() method wont raise any compile time error or runtime exception simply returns false

Q.



```
class Test
{
    public static void main(String[] args)
    {
        String s= new String("Durga");
        StringBuffer sb= new StringBuffer("Durga");
        //System.out.println(s==sb);
        //System.out.println(s.equals(sb));
         System.out.println(s.equals(sb.toString()))
    }
}
```

True because sb.toString return string as result then String.equals(String) it will work as content comparison and give true.