

Hibernate object states-

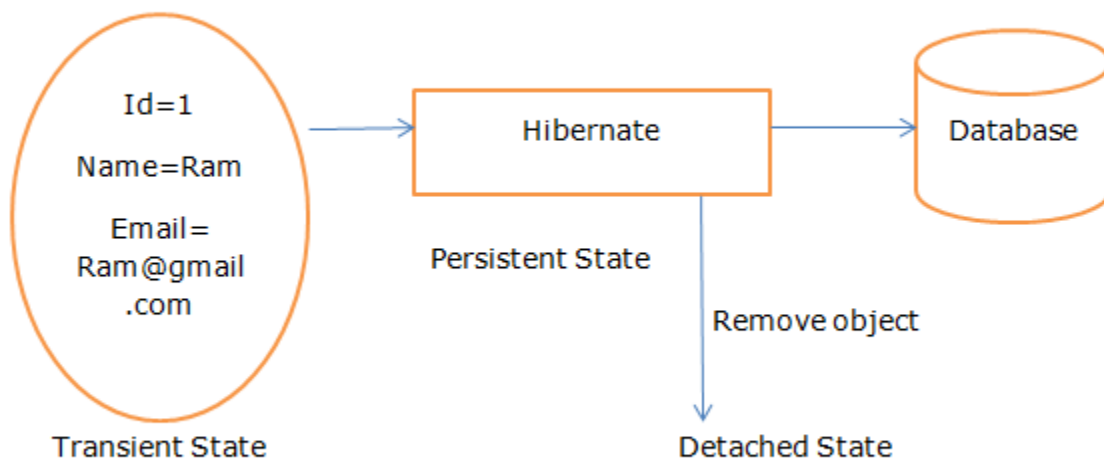
It is called as life cycle of POJO class objects.

There are three types of objects in the hibernate as-

Transient state

Persistent state

Detached state



Transient State-

Simple object or plain object or before going too attached with hibernate called as transient state.

Persistent State-

A plain java old object attached with hibernates called as persistent state.

Detached State-

A process of deleting object from hibernates called as detached state.

How to write the POJO class?

POJO class needs to contain the required number of property which are storing into table.

POJO class in which default constructor is mandatory.

For every property, setter and getter need to be provided.

Setter and getter access should be public.

Note- there are several names for POJO class such as formbean, valueobject, etc

Program-

```
package com.test;
import javax.persistence.Entity;
import javax.persistence.Id;

@Entity
public class Student {

    @Id
    private int id;
    private String name;
    private String city;
    private String mobile;

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
```

```

        this.name = name;
    }
    public String getCity() {
        return city;
    }
    public void setCity(String city) {
        this.city = city;
    }
    public String getMobile() {
        return mobile;
    }
    public void setMobile(String mobile) {
        this.mobile = mobile;
    }
}

```

Test.java

```

package com.test;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.cfg.Configuration;

public class Client {

    public static void main(String[] args) {

        Student student= new Student();
        student.setName("Jeevan");
        student.setCity("Pune");
        student.setMobile("8888978585");
        //student object is transient
        Configuration configuration= new Configuration();
        configuration.configure("hibernate.cfg.xml");

        SessionFactory
sessionFactory=configuration.buildSessionFactory();
    }
}

```

```
Session session=sessionFactory.openSession();
session.save(student);
//student object is persistent

session.beginTransaction().commit();

session.evict(student);
/*student object is detached or student object is
removed from
*persistent state.
* */
    }
}
```