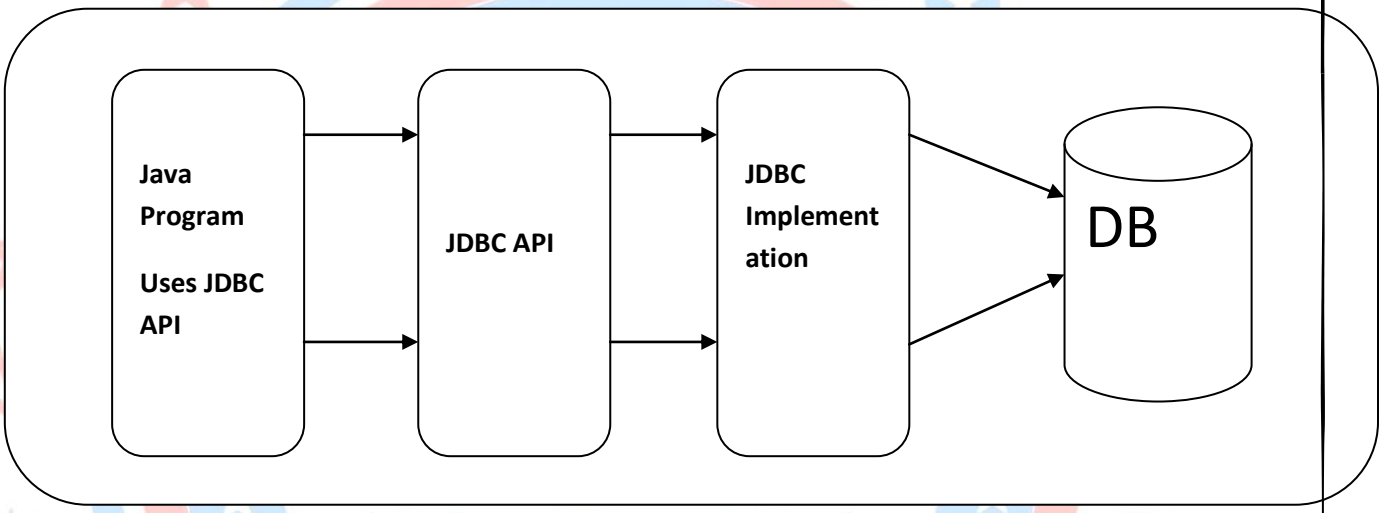
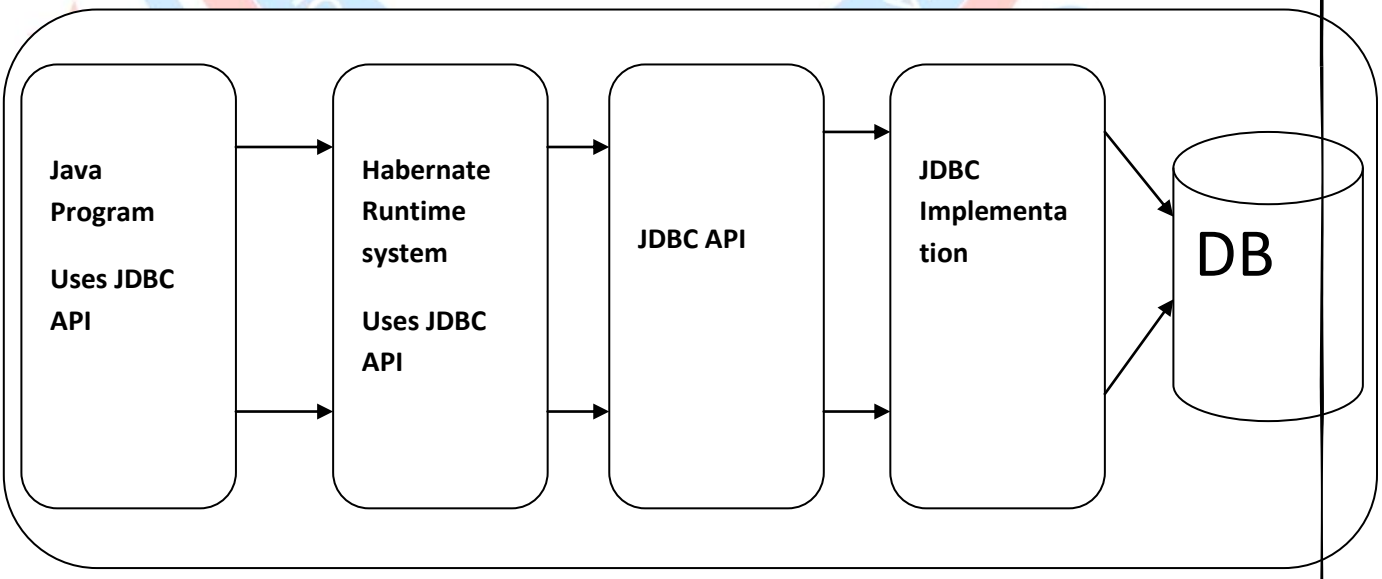
Introduction`

- Hibernate is a persistence framework which is used to implement persistence operations or database operations.
- Hibernate framework was implemented by red hat.
- Architect of Hibernate is Gavin king.
- Hibernate is ORM (object relational mapping) tool and is best among all other persistence frameworks like Abates, top link, JDO etc.
- Hibernate can be implemented either with XML approach (Hibernate core) or annotation approach (Hibernate Annotations).
- JDBC specification is given by SUN and is implemented by various by various database vendors.



- **Hibernate is implemented on the top of JDBC technology i.e. you can use Hibernate API which internally uses JDBC to contact database.**



- When you write the JDBC code for implementing database operations, you need to perform the following:

  Try {

  1. Load the Driver Class
  2. Get the connection
  3. Prepare the SQL statement (***)
  4. Create the JDBC statement
  5. Submit the SQL statement to DB.
  6. Process the results (***)

  } catch (Exception e) {}

  Finally {

  7. Clean the resources

- In the above code, more statements (1, 2, 4, 5, and 7) are common across multiple JDBC programs. This gives you the code duplication problem.
- In JDBC java Developer is responsible to:

1. Generate primary keys.
2. Write SQL statements.
3. Clean the resources.
4. Process the resources.
5. Process the results.
6. Implement batch up dates.

   Etc.

Hibernate Features

1) Hibernate system is responsible for taking the connections, creating statements and releasing the resources.

   EX.

   Customer cust = New customer ("sti"," som@ Jtc", 1234)
   Session save (cust);

   Customer cust = session.load (customer. class, 99)
2) Hibernate system is responsible for generating SQL queries which are well-tuned in terms of performance.
3) Hibernate system is responsible for generating the value required for primary key columns.
4) Hibernate provides many built-in primary key generation algorithms and also supports to implement custom primary key generation algorithms.

5) Hibernate supports various mapping styles:
1. Simple Mapping
2. Collection Mapping
3. Inheritance Mapping
   a. Table per sub class mapping
   b. Table per class mapping
   c. Table per concrete class mapping
4. Association mapping
   a. One-to-one mapping
   b. One-to-many mapping
   c. Many-to-many mapping
5. Other mappings.

6) Hibernate supports two ways to manage connections
a. Driver Manager connections
b. Data Source connections(*)

7) Hibernate supports two ways to manage Transactions
a. JDBC Transactions
b. JTA transactions

8) Hibernate has in built support for Batch Up dates.

9) Hibernate provides various caching mechanism.

10) Hibernate provides various Query languages:
a. HQL(*) Hibernate Query Language
b. QBC(*)Query By Criteria
c. QBE
d. Native SQL
e. Named SQL

11) Hibernate system uses many persistent best practices and forces the developer to use them for better performance.

Steps to develop first Hibernate core example in eclipse

1) Create the java project with the name : Jtc1
2) Add all Hibernate core (3.1) jars to project build path.
3) Copy Hibernate.cfg.xml to src folder.
   Note: Hibernate.cfg.xml is called Hibernate configuration document where you can specifying the database details.
4) Consider the table called customers:

```
DROP DATABASE Jtcindia db;
CREATE DATABASE Jtcindia db;
USE Jtcindia db;

CREATE TABLE customers (
```

Cid Int primary key auto increment, cname VARCHAR (15)
Email VARCHAR (15), phone VARCHAR (15),
City VARCHAR (15), BAL double,

Create a package called com. Jtcindia. Hibernate and write the following:
Hibernate persistence class i.e. customer.java
Hibernate mapping document i.e. customer. Hbm.xml to map your table with persistence class.
Write the following Hibernate client code under package com.jtcindia.Hibernate.

Jtc1A.java
Jtc1B.java

Jtc1: Files required

Jtc1A.java	Jtc1B.java	Customer.java
Customer. hbm.xml	Hibernate. cgf.xml	CHibernateUtil.java

JTC1:

```
==================
package com.jtcicindia.hibernate;
import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
public class Jtc1A{
public static void main(string[]args){
Transaction tx=null;
try{
SessionFactory
sf=CHibernateUtil.getSessionFactory();
Session session=sf.openSession();
tx=session.beginTransaction();
Customer cust=new
customer("som",som@jtc",123,"Noida",25000);
session.save(cust);
tx.commit();
session.close();
System.out.println("record inserted");
)catch(Exception e){
e.prinStackTrace();
if(txl=null)tx.rollback();
}
}}
```

2.Jtc1B.java

```
==================
package com.jtcicindia.hibernate;
import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
public class Jtc2B{
public static void main(string[]args){
Transaction tx=null;
try{
SessionFactory
sf=CHibernateUtil.getSessionFactory();
Session session=sf.openSession();
tx=session.beginTransaction();

Customer cust =(Customer)
session.load(Customer.class,1);
system.out.println(cust.getCld()+"/t"+cust.getCname()
+"\t"+cust.getEmail()+cust.geatphone());

tx.commit();
}catch(Exception e){
e.printStackTrace();
if(tx!=null)tx.rollback();
}
```

3.Customer.java
==========
```
package com.jtcindia.hibernate;
public class Customer{
Private int cid;          //P.K
private String cname;
private string email;
private long phone;
private string city;
private double bal;

public Customer() {}
Public customer(string cname, string email, long phone,
String city, double bal)(
this.cname=cname;
this.email=email;
this.phone=phone;
this.city=city;
this.bal=bal;
}
        //Setters and getter
}
```
4.customer.hbm.xml
==========
```
<hibernate-mapping
package="com.jtcindia.hibernate">
<class name="customer" table="customer">
<id name="cid"column="cid"type="int">
<generator class="increment"/>
</id>
<property name="cname"/>
<property name="email"column="email"/>
<property
name="phone"cloumn=phone"type="long"/>
<property
name="city"column="city"type="string"/>
<property
name="bal"column="bal"type="double"/>
</class>
</hibernate-mapping>
```

}}
5.hibernate.cfg.xml
==========
```
<hibernate-configuration>
<session-factory>
<property name="hibernate.connection.driver_class">
        com.mysql.jdbc.Driver
</property>
<propety name="hibernate.connection.url">
        jdbc:mysql://localhost:3306/jtcindiadb
</property>
<property name="hibernate.connection.username">
        root
</property>
<property name="hibernate.connection.password">
        somsree
</property>
<property name="dialect">
        org.hibernate.dialect.MySQLDialect
</property>
<property name="show_sql">true</property>
<mapping
resource="com/jtcindia/hibernate/customer.hbm.xml/>
</session-factory>
</hibernate-configuration>
```
6.cHibernateUtil.java
==========
```
package com.jtcindia.hibernate;
import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
public class CHibernateUtil{
static SessionFactory factory;
static{
try{
Configuration cfg=new Configuration();
cfg=cfg.configure();
factory=cfg.buildSessionFactory();
}catch(Exception e){e.printStacktrace();}
}
Public static SessionFactory getSessionFactory(){
return factory;}}
```

Steps to develop First Hibernate Annotation Example in Eclipse:

1. Create the java project with the name: Jtc2
2. Add all Hibernate Annotation (3.2) jars to project to project build path.
3. Copy Hibernate. Cfg. xml to src folder.
4. Note: Hibernate. Cfg. xml is called Hibernate Configuration document where you can specifying the database details.
5. Consider the table called customers
6. DROP DATABASE Jtcindia db;
7. CREATE DATABASE Jtcindia db;
8. USE Jtcindia db;
9. CREATE TABAS customer (
   Cid Int primary key auto increment, cname VARCHAR (15)
   Email VARCHAR (15), phone VARCHAR (15)
   City VARCHAR (15), BAL double);
10. Create a package called com. Jtcindia. Hibernate and write the following:
11. Hibernate persistence class i.e. customer.java
12. Write the following Hibernate client code under package com. Jtcindia. Hibernate.

Jtc1A.java
Jtc1B.java

Jtc2: Files required

Jtc1A.java	Jtc1B.java	Customer.java
Hibernate. cgf.xml	AHibernateUtil.java	

1.Jtc1B.java

```
==================
package com.jtcicindia.hibernate;
import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
public class Jtc2B{
public static void main(string[]args){
Transaction tx=null;
try{
SessionFactory
sf=AhibernateUtil.getSessionFactory();
  //Same as Jtc1A.java
}catch(Exception e){
e.printStackTrace();
if(tx!=null)tx.rollback();}}}
```

2.Jtc1B.java

```
==================
package com.jtcicindia.hibernate;
import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
public class Jtc2B{
public static void main(string[]args){
Transaction tx=null;
try{
SessionFactory
sf=AhibernateUtil.getSessionFactory();
  //Same as Jtc2A.java
}catch(Exception e){
e.printStackTrace();
if(tx!=null)tx.rollback();}}}
```

3.Customer.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
```

4.hibernate.cfg.xml

```
======================
<hibernate-configuration>
```

```
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
@Entity
@table(name="customers")
Public class customer{
@id
@GeneratedValue(strategy=GenerationType.AUTO)
@Column(name="cid")
Private int cid;//P.K
@Column(name="cname")
Private string cname;

@Column(name="email")
Private string cname;

@Column(name="phone")
Private string cname;

@Column(name="city")
Private string cname;

@Column(name="bal")
Private string cname;

Public customer() {}
Public Customer(String cname,String email,long
phone,String city, double bal){
this.cname=cname;
this.email=email;
this.phone=phone;
this city=city;
this.bal=bal;
}
        //Setters and Getters
}
```

```
<session-facttory>
<property
name="hibernate.connection.driver_class">
        Com.mysql.jdbc.Driver
</property>
<property name="hibernate.connection.url">
         Jdbc:mysql://localhost:3306/jtcindiadb
</property>
<property name="hibernate.connection.username">
        Root
</property>
<property name="hibernate.connection.password">
    somsree
</property>
<property name="dialect">
        Org.hibernate.dialect.MySQLDialect
</property>
<pcoperty name="show_sql">true</property>
<mapping
class="com.jtcindia.hibernate.Customer"/>
</session-factory>
</hibernate-configuration>
```

5.AhibernateUtil.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
Import org.hibernate.cfg.*;
/*
*@author:Som Prakash Rai
*@comapany:Java Training Center
*@See          :www.jtcindia.org
**/
Public class AhibernateUtil{
Static sessionFactory factory;
Static{
AnnotationConfiguration cfg=new
annotationConfiguration();
cfg=(AnnotationConfiguration)cfg.configure();
factory=cfg.buildSessionFactory();
}
Public static SessionFactory setSessionFactory() {
Return factory;
}
```

Questions

Q1) how many instances will be created for the persistence class with the following client code with hibernate core?

Ans: As per above code, 0 objects will be created.

Q2) how many instances will be created for the persistence class with the following client code with Hibernate core?

```
Configuration Cfg=new Configuration ();

Cfg=Cfg. Configure ();

Session factory =Cfg. Build session factory ();
```

Ans: 3 objects will be created using default constructor for the reflection process at the time of creating the session factory object.

```
Configuration Cfg = new configuration ();
Cfg=Cfg. Configure ();
Session factory = Cfg. Build session factory ();
Session =factory. OpenSession ();
TX=session. BeginTransaction ();
Customer c1=new customer ("sp"," sp@ jtc", 1234,"noida");
Session.  Save (c1);
Customer c2= new customer ("sp"," sp@ jtc", 1234,"noida");
Session. Save (c2);
…….
```

Ans: As per above code, 2 objects will be created with argument constructor but at the time of creating the Session factory object, 3 objects will be created using default constructor for the reflection process.

Q4) how many instances will be created for the persistence class with the following client code with Hibernate Annotation?

```
AnnotationConfiguration Cfg = new AnnotationConfiguration ();

Cfg = (AnnotationConfiguration) Cfg. Configure ();

Session factory =Cfg. Build session factory ();
```

Ans: 1 object will be created using default constructor for the reflection process at the time of creating the session factory object.

Q5) how many instances will be created for the persistence class with the following client code with hibernate Annotation?

```
Annotation Configuration Cfg = new annotation configuration ();
Cfg= (annotation configuration) Cfg. Configure ();
Session factory = Cfg. Build session factory ();
Session =factory. OpenSession ();
Tax=session.BeginTransaction ();
Customer c1=new customer (“sp”,” sp@ jtc”, 1234,”noida”);
Session. Save (c1);
Customer c2= new customer (“sp”,” sp@ jtc”, 1234,”noida”);
Session. Save (c2);
…….
```

Ans: As per above code, 2 objects will be created with argument constructor but at the time of creating the session factory object, 1 object will be created using default constructor for the reflection process.

Q6) what will happen when I specify the field name wrongly with mapping document or annotation?

Ans: You will get an exception org. hibernate. Property not found exception: field not found: cname 1

Q7) what will happen when I specify the column name wrongly with mapping document or annotation?

Ans: Unknown column cname1 in field list.

- **Hibernate supports various mapping styles**:
  1. Simple mapping
  2. Collection mapping
  3. Inheritance mapping
     a. Table per sub class mapping
     b. Table per class mapping
     c. Table per concrete class mapping
  4. Association mapping
     a. One-to-one mapping
     b. One-to-Many mapping
     c. Many-to-many mapping
  5. Other mappings.

Simple mapping

- When you map your Hibernate persistence class with simple data types like string, primitives, wrappers, data etc with the corresponding table columns then it is called as simple mapping.
  Ex:

```
Public class customer {
Private Int cid;
Private string cname:
Private string email;
…..
}
```

With xml:

- For primary key property, use <id> or <composite-id> tag.
  Ex:
  <id name=”cid” column=”cid” type=”Int”>
- For the simple properties, use <property> tag.

  <property name=”cname”/>

  <property name=”email “column=”email”/>

  <property name=”phone “column=”phone “type=”long”/>

With Annotation:

- You need to use following annotations for persistence class
  @Entity
  @Table (name=”customers”)
  Class customer {……}
- You need to use following Annotations for primary key field
  @Id
  @column (name=”cid”)
  Private Int cid;
- You need to use following Annotations for other simple fields
  @Column (name=”cname”)
  Private string cname:

Collection mapping

When you map your Hibernate persistence class with collection data types like array, list, set, map then you have to use collection mapping.

Example using Hibernate core (Jtc3):

A) Persistence class

```
Class student {
Int sid;//P.K
String sname;
String DOB;
String qualification;
String [] courses;
List<string>emails;
List<Integer> marks;
Set<long> phones;
Map<string, long>refs;
……..
}
```

My students

Sid	Sname	dob	Qualification
101	Som	25-12-1989	M.Sc

Courses

Sid	Cname	Idx
101	Java	0
101	JDBC	1
101	JSP	2

Refs		
Sid	R name	R phone
101	Aaa	1111
101	Bbb	2222
101	ccc	3333

Emails		
Sid	Emailed	Idx
101	aa@jtc.org	0
101	bb@jtc.org	1
101	cc@jtc.org	2

Marks

Sid	Marks
101	100
101	99
101	100

D) Client Code

```
String[]crs={"Java","JDBC","JSP"};
List<string>ems=new ArrayList<String>();
```

```
        Ems.add("aa@jtc");
        …
        List<Integer>mks=new ArrayList<Integer>();
        mks.add(100);
        ….
        Set<Long>phs=new HashSet<Long>();
        Phs.add(new Long(1111);
        …..
        Map<String,long>refs=new HashMap<string,Long>();
        Refs.put("AAA",new Long(111);
        ….
        Student stu=new Student("Som","10-10-10","M.Sc",crs,ems,mks,phs,refs);
        Session.sava(stu);
```

D) Mapping Document

```
        <hibernate-mapping package="com.jtcindia.hibernate">
        <class name="student"table="mystudent">
        <id name="sid"type="int"column="sid">
        Generateor class="increment"/>
        <id>
<property name="sname"/>
<property name="dob"/>
<property name="qualification"column=quall"/>
        <array name="courses"table=courses">
                <key clolumn="sid"/>
                <index column="idx"/>
                <element column="cname"type="string"/>
        </array>
        <list name="emails"table="emails">
                <key clolumn="sid"/>
                <index column="idx"/>
                <element column="cname"type="string"/>
        </list>
        <bag name="marks"table=table="marks">
                <key clolumn="sid"/>
                <element column="marks"type="int"/>
        <bag>
        <3ct name-"phones"table="phones">
                <key clolumn="sid"/>
                <element column="phone"type="long"/>
        </set>
        <map name="refs"table="refs">
                <key clolumn="sid"/>
                <index column="rphone"type="string"/>
                <element column="rphone"type="long"/>
        </map>          </class> </hibernate-mapping>
```

Jtc3A.java	Jtc3B.jav a	Student.java
Student.hbm.xml	Hibernate.cgf.xml	ChibernateUtil.java

Jtc3A.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class Jtc3A {
Public static void main (string[]args){
Try{
SessionFactory sf=ChibernateUtil.getFactory();
Session session=sf.openSession();
Transaction tx=session.beginTransaction();
String cous[]={“java”,”JDBC”,”JSP”];
List<string>ems=new ArrayList<string>();
Ems.add(“aa@jtc”);
Ems.add(“bb@jtc”);
Ems.add(“cc@jtc”);
List<Integer>maks=new ArrayList<Integer>();
Maks.add(new Integer(100));
Maks.add(new Integer(99));
Maks.add(new Integer(100));
Set<Long>phs=new HashSet<long>();
Phs.add(new Long(1111));
Phs.add(new Long(2222));
Phs.add(new Long(3333));
Map<string,Long>refs=new
HashMap<String,Long>();
Refs.put(“aaa”,new Long(1111));
Refs.put(“bbb”,new Long(2222));
Refs.put(“ccc”,new Long(3333));
Student stu=new student(“som”,”10-10-
10”,”M.Sc”,cous,ems,maks,phs,refs);
Session.save(stu);     Tx.commit();
Session.close();
}catch(Exception e){e.prinstStackTrace();
}          }          }
```

Jtc3B.jav a

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class Jtc3A {
Public static void main (string[]args){
Try{
SessionFactory sf=ChibernateUtil.getFactory();
Session session=sf.openSession();
Transaction tx=session.beginTransaction();
Student stu=(Student)session.load(student.class,1)
System.out.println(stu.getSid()+”\t”+stu.getSname()
+”\” t+stu.getDob()){
Ststem.out.println(cn);
}
System.out.println(sty,get/enauks());
System.out.println(sty,get/Marks());
System.out.println(sty,get/phones());
System.out.println(sty,get/refs());
Tx.commit();
Session.close();
}catch(Exception e){
e.printStackTracke();
}}}
```

Student.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training Center
*@See        : www.jtcindia.org
**/
Public class Student{
Private int sid;//P.K
Private string sname;
Privatge stringdob;
Private string qualification;
Private string[] courses;
Private list<String>emails;
PrivateList<Integer>marks;
Private set<Long>phones;
Private map<String,Long>refs;
Public Student(){}
Public Student(String sname,string dob,
stringqualifaction,
String[]courses,
List<String>emails,List<Integer>Marks,
Set<long>phones,map<string,Long>regs)(super();
This.sname = sname;
This.dob =dob;
This.qualification=qualification;
This.courses = courses;
This.emails =marks;
This.phones=phones;
This.refs = refs;
}
//setters and getters
}
```

Hibernate.cfg.xml

```
<hibernate-configuration>
<session-factory>
…….
<property
name=”hbm2ddl.auto”>update</property>
<maping
resource=”com/jtcindia/hibernate/student.hbm.xml”/>
</session-factory>
</hibernate-configuration>
```

Student.hbm.xml

```
<hibernate-mapping
package=”com.jtcindia.hibernate”>
<class name=”student”table=”students”>
<id name=”sid”column=”sid”type=”int”>
<generator class=”increment”/>
</id>
<property name=”sname”/>
<property name=”dob”/>
<property name=”qualification”column=”quali/>
<array name=”courses”table=courses”>
<key column=”sid”/>
<index column=”idx”/>
<element column=”cname”type=”string/>
</array>
<list name=”emails”table=”emails”>
<key column=”sid”/>
<index column=”idx”/>
<element column=”marks”type=”string”/>
</list>
<bag name=”marks”table=”marks”>
<key column=”sid”/>
<element column=”marks”type=”int”/>
</bag>
<set name=”phones”table=phones”.
<key column=”sid”/>
<element column=”phoneNo”type=”long”/>
</set>
<map name=”refs”table=”refs”>
<key column=”rphone”type=”long”/>
</map>
</class>
</hibernate-mapping>
```

Example using Hibernate Annotation (Jtc4);
Persistence Class

```
@Entity
@Table(name=”MyStudents”)
Public class Student(
@Id
@Generated Value(strategy=GenerationType.AUTO)
@Column(name=”sid”)
Private int sid;//P.K

@ColumnI(name=”sname”)
Private string sname;

@column(name=”dob”)
Private string dob;

@column(name=”quali”)
Private string qualification;

@collectionOfElements
@Join Table(name-courses”,joinColumns=@joinColumn(name=”sid))
@IndexColumn(name=”idx”)
@column(name=”cname”)
Private string[]courses;

@collectionOfElements
@Join Table(name-courses”,joinColumns=@joinColumn(name=”sid))
@IndexColumn(name=”idx”)
@column(name=”cname”)
Private List<String>emails;

@collectionOfElements
@Join Table(name-courses”,joinColumns=@joinColumn(name=”sid))
@IndexColumn(name=”idx”)
@column(name=”cname”)
Private List<Integer>markes;

@collectionOfElements
@Join Table(name-courses”,joinColumns=@joinColumn(name=”sid))
@IndexColumn(name=”idx”)
@column(name=”cname”)
Private set<Long>phones;

@collectionOfElements
@Join Table(name-courses”,joinColumns=@joinColumn(name=”sid))
@IndexColumn(name=”idx”)
@column(name=”cname”)
```

Private Map<string,Long>refs;

B) Tables required

My students			
Sid	Sname	Dob	Qualification
101	Som	25.12.1989	M.SC

	Courses	
Sid	Cname	Idx
101	Java	0
101	JDBC	1
101	JSP	2

	Emails	
Sid	Emailed	Idx
101	aa@jtc.org	0
101	bb@jtc.org	1
101	cc@jtc.org	2

Marks

Sid	Marks
101	100
101	99
101	100

	refs	
Sid	Mapkey	Rphone
101	Aaa	1111
101	Bbb	2222
101	Ccc	3333

Phones

Sid	phone
101	1111
101	2222
101	3333

Specifying the Tabe	Table name has to be specifed which are reqresenting collections called-> <array>,<list>,<bag>,<set>,<map> Usage; <list….table="emails">	Table name has to be specified with @join Table Usage; @Join Table (name="emails")
Specifying the F.K	<key column="sid">	@join Column(name="sid")
Apecifying Index column	<index column="idx"/>	@Index column(name="idx")
Specifying the Elem ent column	<element column="emailld" type="string"/>	@Column(name="emailld")

Jtc4:Files required

Jtc4A.java	**JtcB.java**	**Student.java**
Hibernate.cgf.xml	**AhibernateUtil.java**	

Jtc4A.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class Jtc4A{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session
Factory();
//Same as Jtc3A.java
}catch(exception e){
e.printStack Trace();
if(tx!=null)tx.rollback();
}
}
```

Jtc4B.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class Jtc3B{
Public static void main(string[]args){
Transaction tx=null;
Try{
sessionFactory
sf=AhibernateUtill.gettSessionFactory();
//Same as Jtc3B.java
}cetch(Exception e){
e.prinStack Tarace();
if(tx!=null)tx.rollback();
}}}
```

Hibernate.cfg.xml

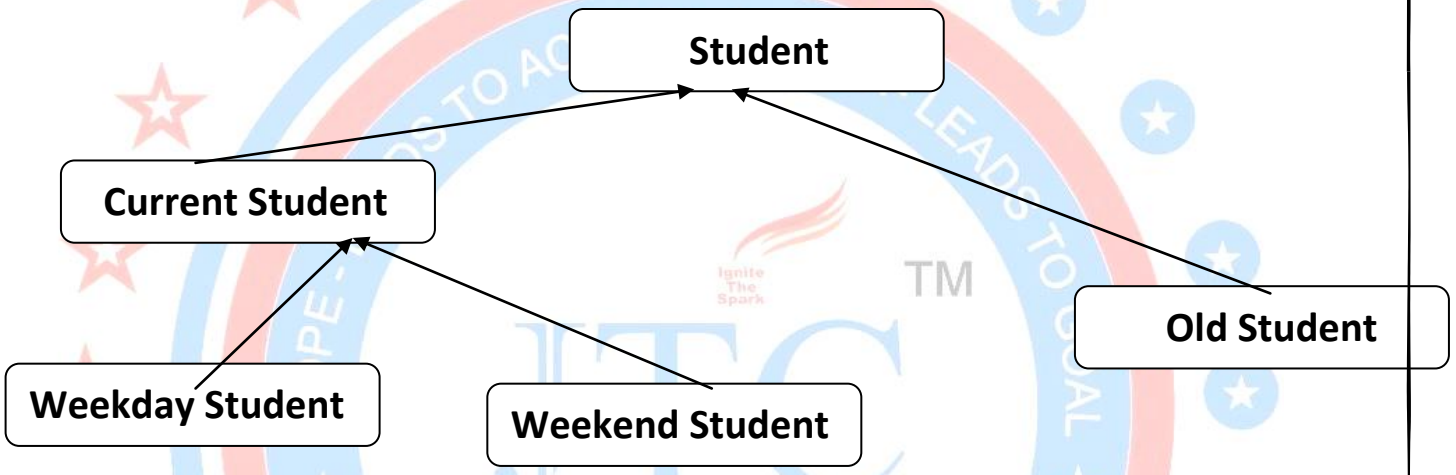
Student.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
@Entity
@Table(name=”MyStudents”)
Public class Student(
@Id
@Generated
Value(strategy=GenerationType.AUTO)
@Column(name=”sid”)
Private int sid;//P.K

@ColumnI(name=”sname”)
Private string sname;

@column(name=”dob”)
Private string dob;

@column(name=”quali”)
Private string qualification;

@collectionOfElements
@Join Table(name-
courses”,joinColumns=@joinColumn(name=”sid)
)
@IndexColumn(name=”idx”)
@column(name=”cname”)
Private string[]courses;

@collectionOfElements
@Join Table(name-
courses”,joinColumns=@joinColumn(name=”sid)
)
@IndexColumn(name=”idx”)
@column(name=”cname”)
Private List<String>emails;

@collectionOfElements
@Join Table(name-
courses”,joinColumns=@joinColumn(name=”sid)
)
@IndexColumn(name=”idx”)
```

<hibernate-configuration>
<session-factory>
…….
<property name=”hbm2ddl.auto”>update</property>
<maping resource=”com/jtcindia/hibernate/student.hbm.xml”/>
</session-factory>
</hibernate-configuration>
	@column(name=”cname”)
	Private List<Integer>markes;
	
	@collectionOfElements
	@Join Table(name-courses”,joinColumns=@joinColumn(name=”sid”)
	)
	@IndexColumn(name=”idx”)
	@column(name=”cname”)
	Private set<Long>phones;
	
	@collectionOfElements
	@Join Table(name-courses”,joinColumns=@joinColumn(name=”sid”)
	)
	@IndexColumn(name=”idx”)
	@column(name=”cname”)
	Private Map<string,Long>refs;
	
	Public Student(){}
	Public Student(String sname,string dob,
	stringqualifaction,
	String[]courses,
	List<String>emails,List<Integer>Marks,
	Set<long>phones,map<string,Long>regs)(super()
	;
	This.sname = sname;
	This.dob =dob;
	This.qualification=qualification;
	This.courses = courses;
	This.emails =marks;
	This.phones=phones;
	This.refs = refs;
	}
	//setters and getters
	}

Inheritance Mapping

- When multiple jape entities are in inheritance relationship then use Inheritance mapping.
- You can implement Inheritance mapping in 3 ways
  1. Table per sub class mapping
  2. Table per class mapping
  3. Table per concrete class mapping
- **Consider the following JPA Entities with inheritance relationship.**



1) Table per sub class mapping

- In this mapping you need to take one table per one sub class.
- Student is the main super class which will have the master table called my students.
- Every subclass will have its own table.
- When you save student class object then
  - Only one record will be inserted in my students.
- When you save current student class object then.
  - One record will be inserted in my students.
  - One record will be inserted in students
- When you save Old Student class object then.
  - One record will be inserted in my students.
  - One record will be inserted in ostudents.
- When you save weekday student class object then
  - One record will be inserted in my students.
  - One record will be inserted in csteudents.
  - One record will be inserted in wdstudents.

- When you save weekend student class object then
  - One record will be inserted in my students.
  - One record will be inserted in csteudents.
  - One record will be inserted in wdstudents.

A) Tables required

1) My students

Sid	Sname	City	Status	total fee

2) Csteudents

Sid	Feebal	Timings	Branch

3) Ostudents

Sid	Oscompany	Osemail	Osctc

4) Wdstudents

Sid	Qualification	Percentage	yop

5) Westudents

Sid	Wecompany	Weemail	wectc

Example with Hibernate core (Jtc5)
Hibernate Persistence classes

```
Public class student{
Private Int Sid;
Private string sname;
Private string city;
Private string status;
Private double total fee;
…..
}
```

2. Current student.java

```
Public class current student extends student {
Private double Feebal;
Private string timings;
Private string branch;
}
```

Old student.java

```
Public class Old student extends student {
Private String Company;
Private double octc;
Private string ocemail;
}
```

Weekday student.java

Public class Weekday student extends student {
Private string ocompany;
Private string percentage;
Private Int you;
}

Weekend student.java

Public class Weekend student extends student {
Private string wcompany;
Private string wcemail;
Private double wctc;
}

Mapping document

Jtc5: Files required

Jtc5a.java	Jtc5B.java
Student. java	Current student.java
Old student. java	Weekday student.java
Weekend Student.java	CHibernate Util.java
Student. Hbm. xml	Hibernate. Cfg. xml

Jtc5A.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class Jtc5A{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session
getFactory();
Session session=factory.openSession();
Tx=session.beginTransaction();

//1.Adding the student
Student stuenew
Stuednt("som","Noida","Enabled",15000.0);
Integer it=(integer)session.save(stu);
System.out.println(it.intValue());

//2.Adding the CurrenStudent
Current Student cstu=new
CurrentStudent("Rai","Noida","Enabled",15000
.0, 2000.0 ,"6.30P.M","Sec-2");
It=(integer)session.save(cstu);
System.out.println(it.intValue());

//3.Adding the oldStudent
OldStudent ostu=new
OldStudent
("aa","Noida","Enabled",15000.0,"WEBEARN
", "aa@sp.com9.0);
It=(integer)session.save(ostu);
System.out.printin(it.intValue());

//4.Adding the regular student
WeekdayStudent Wdstu=new
regularStudent('bb','noida","Enabled",15000.0,
2000.0"6.30P.M","M.sc","85.5",3);
it=(Integer)session.save(wdstu);
system.out.printin(it.intValue());

//5.Adding the WeekendStudent
WeekendStudent wstu=new
weekendStudent("cc","Noida","Enabled",15000.0,
2000.0,
6.30P.M","GRNOIDA"<"WEBEARN",cc@sp.com
,9.0);
it =(integer)session.save(wstu);
System.out.println(sty,get/refs());
Tx.commit();
Session.close();
}catch(Exception e){
e.printStackTracke();
if(tx!=null)tx.rollback();
}
}
}
```

Jtc5B.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class Jtc3A {
Public static void main (string[]args){
Try{
SessionFactory sf=ChibernateUtil.getFactory();
Session session=sf.openSession();
tx=session.beginTransaction();
System.(string)session.load(student.class.1);
System.out.println(st);
Tx.commit();
Session.close();
}catch(Exception e){
e.printStackTracke();
if(tx!=null)tx.rollback();
}
}
}
```

Student.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class student{
Private int sid;            private string sname;
Private string city;     private string status;
Private double totalfee;
Publice student() {}
Public student(string sname,string city, string
status,double totalfee)(
This.sname=sname;
This city=city;
This.status status;
This.totalfee=totalfee;
}
//setters and getters
@override
Public string to string(){
Return
sid+”,”+sname+”,”+city+”,”+status+”,”+totalfe
e;
}
}
```

CurrentStudent.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class CurrentStudent extends student{
Private int feebal;          private string sname;
Private string branch;
Publice CurrentStudent() {}
Public CurrentStudent(string sname,string city,
string status,double feebal,stringtimings,string
brach)(
Super(sname,city,status,totalfee);
This feebal= feebal;
This.timings=timings;
This.branch=branch;
}
//setters and getters
@override
Public string to string(){
Return super.to string()
+”,”+feebal+”,”+timings+”,”+branch;
}
}
```

OldStudent.java	WeekdayStudent.java

Package com.jtcindia.hibernate;
Public class oldStudent extends student{
Private string occompany;
Private string ocemail;
Privat double octc;
Public OldStudent(){}
Public oldStudent{string sname,string city, string status, double totalfee, string ocompany, string ocemail, double octc){
Super(sname,city, status, totalfee);
This.ocompany=ocompany; this.ocemail = ocemail;
This.octc=octc;
}
//Setters and getters
Public string to string(){
Return super.to string()+","+company+email+email+","+ctc;
}
	Package com.jtcindia.hibernate;
	Public class weekdayStudent extends CurrentStudent{
	Private string qualification;
	Private string percentage;
	Privat double yoe;
	Public WeekdayStudent(){}
	Public WeekdayStudent{string sname,string city, string status, double totalfee, double feeble, string timings, string branch,string qualification, string percentage, int yoe){
	Super(sname,city, status, totalfee);
	This.qualification= qualification; this.percentage = percentage;
	This.yoe= yoe;
	}
	//Setters and getters
	Public string to string(){
	Return super.to string()+","+qualification +","+top;
	}

WeekendStudent.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training center
*@See   :Www.jtcindia.org
*@call :9990399111
**/
Public class weekend Student extends
currentStudent{
Private string wcompany;
Private string wcemail;
Private double wctc;
Public weekendstudent(){}
Public weekendString sname, string city, string
status, double totalfee, double feebal, string timings,
string branch, string wcompany, string wcemail,
double wctc){
Super(sname,city, status, totalfee,feebal, timings,
branch);
This wcompany=wcompany;
This.wcemail=wcemail;
This.wctc=wctc;
}
//setters and getterrs
@override
Public string to string() {
Return super. To
string()+","+company+email+","+etc;
}
}
```

Hibernate.cfg.xml

```
<hibernate-configuration>
<session-factory>
…….
<mapping
resource="com/jtcindia/hibernate/student.hbm.xml"/>
</session-factory>
</hibernate-configuration>
```

Student.hbm.xml

```
<hibernate-mapping
package="com.jtcindia.hibernate">
<class name="student"table="mystudents">
<id name="sid"column="sid"type="int">
<generator class="lncrement"/>
</id>
<property name="sname"/>
<propety name="city"/>
<propety name="status"/>
<propety name="totalfee"type="double"/>
<joined-subclass name="currentStudent"
Table"cstudents">
<key column="sid"/>
<property name="feebal"type="type"double"/>
<property name="timings"/>
<property name="branch"/>
<joined-subclass name="weekdayStudent"
Table="wdstudents">
<key column="sid"/>
<property name="qualification"/>
<property name="percentage"/>
<property name="yoe"type="int"/>
</joined-subclass>
<joined-subclass name="weekendStudent"
Table="westudents">
<key column="sid"/>
<property neme="wcompany"/>
<property name="wcemail"/>
<property name="wctc"type="double"/>
<"joined-subclass>
<joined-subclass name="oldStudent"
Table="ostudents">
<key column="ocompany"/>
<property name="ocmpany"/>
<property name="ocemail''/>
<property name="octc"type="double"/>
</class>
</hibernate-mapping>
```

Example with hibernate annotation (Jtc6)

B) Hibernate persistence classes

1. Student.java

```
@Entity
@ Table (name=”my students”)
@Inheritance (strategy=Inheritance Type. JOINED)
Public class student {
…..
}
```

2. Current student.java

```
@Entity
@ Table (name=”customers”)
@Primary key join column (name=”sid”)
Public class current student extends student {
……
}
```

3 .Old student.java

```
@Entity
@ Table (name=”ostudents”)
@Primary key join column (name=”sid”)
Public class old student extends student {
……
}
```

4 .Weekday student.java

```
@Entity
@ Table (name=”wdstudents”)
@Primary key join column (name=”sid”)
Public class weekday student extends student {
……
}
```

Note:

- In the case of Table per sub class mapping,
  - Use the following for super class
    @ Inheritance (strategy = Inheritance Type. JOINED)
  - Use the following for all sub classes
    @ Primary key join column (name = "sid")

Jtc6: Files required

Jtc5a.java	Jtc5B.java
Student. java	Current student.java
Old student. java	Weekday student.java
Weekend Student.java	A Hibernate Util.java
Hibernate. Cfg. xml	

Jtc6A.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training Center
*@See        : www.jtcindia.org
**/
Public class Jtc15A{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session
Factory();

// Same as Jtc5A

}catch(Exception e){
e.printStackTrace();
tx.rollback();
}
}
}
```

Jtc6B.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training Center
*@See        : www.jtcindia.org
**/
Public class Jtc15B{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session
Factory();

// Same as Jtc5A

}catch(Exception e){
e.printStackTrace();
tx.rollback();
}
}
}
```

Student.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”my students”)
@Inheritance(strategy=InheritanceType.JOINED)
Public class student{
@id
@Column(name=”sid”)
@GeneratedValue(strategy=generationType.AUTO)
Private int sid;
@column(name=”sname”)
Private string sname;
@Column(name=”city”)
Private string city
@Column(name=”status”)
Private string status;
@Column(name=”totalfee”)
Private double totalfee;

Public student() {
}

Public student (string sname, string city, string
status, double totalfee) {
This.sname=sname;
This.city=city;
This status = status;
This. Totalfee=totalfee;
}

//setter and getter
@override
Public string to string(){
Return
sid+”,”+sname+”,”+city+”,”+status=”,”+totalfee;
}
}
```

CurrentStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”cstudents”)
@primaryKeyJoinColumn(name=”sid”)
Public class currentStudent extends student{
@column(name=”feebal”)
Private double feebal;
@column(name=”timings”)
Private string timings;
@private string branch;
Private string branch;

Public currentStudent(){
}

Public currentStudent (string sname,string city,
string status,
Double totalfee, double feebal, string timings,
string branch){
Super(sname, city, status, totalfee);
This. Feebal=feebal;
This.timings=timings;
This.branch=branch;
}

//setter and getter
@override
Public string to string(){
+”,”+feebal+”,”+timings+”,”+branch;
}
}
```

WeekendStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”westudents”)
@primary key joinColumn(name=”sid”)
Public class weekendStudent extends
currentStudent{
Private string company;
Private string email;
Private string ctc;

Public weekendStudent() {}

Public weekend Student (string sname, string city,
string status,
Double totalfee, double feebal, string timings, strig
branch,
String company, string email, string ctc){
Super(sname,city, status, totalfee, feebal, timings,
branch);
This.company=company;
This.email=email;
This.ctc=ctc;
}

//Setter and getter

@Override
Public string to string(){
Return super.to
string()+”,”+company+”,”+email+”,”ctc;
}
}
```

Hibernate.cfg.xml

```
<hibernate-configuration>
<session-factory>
…….
<mapping
class=”com.jtcindia.hibernate.student”/>
<mapping
Class=”com.jtcindia.hibernate.currentStudent”/>
<mapping
class=”com.jtcindia.hibernate.oldStudent”/>
<mapping
class=”com.jtcindia.hibernate.oldStudent”/>
</session-factory>
</hibernate-configuration>
```

Table per class mapping

- **In this mapping, you need to take only one table for the entire super and sub classes.**
- **It is also classed as single table mapping.**

A) Tables required

1> my students

Sid	Subtype	Sname	City	Status	Total fee	Feebal	Timings	Branch	Oscompany

Osemail	Osctc	Qualification	Percentage	Yop	Wecompany	Weemail	wectc

Example with Hibernate core (Jtc7)
Hibernate persistence classes

```
1. Public class student {
        Private Int Sid;
        Private string sname;
        Private string city;
        Private string status;
        Private double total fee;
        …..
}
2. Current student.java
Public class current student extends student {
Private double Feebal;
Private string timings;
Private string branch;
}
3. Old student.java
Public class Old student extends student {
Private String Company;
Private string ocemail;
Private double octc;

        }

Weekday student.java
Public class Weekday student extends Current student {
Private string qualification;
Private string percentage;
Private Int you;
        }
Weekend student.java
```

Public class Weekday student extends Current student {
Private string wcompany;
Private string wcemail;
Private double wctc;
}

B) Mapping Document

```
<hibernate-mapping package="com.jtcindia.hibernate">
    <class name="Student"table="mystudents">
        <id name="sid"column="sid"type="int">
            <generator coass="increment"/>
        </id>
        <discriminator column="stutype"/>
        <property name="city"/>
        <property name="city"/>
        <property name="status"/>
        <property name="totalfee"type=double"/>
        <subclass name="currentStudent"discrimintor-value="CSTU">
    <property name=feebal"type="double"/>
        <property name="timing"/>
        <property name="branch"/>
    <subclass name="currentStudent"discrimintor-value=" RSTU
        <property name="qualification"/>
        <property name="percentage"/>
        <property name="yoe"type="int"/>
    </joined-subclass>
    <subclass name="currentStudent"discrimintor-value="WSTU">
        <property name=" wcompany"/>
        <property name="  wcemail"/>
    <property name="  wctc"type="double"/>
        </joined-subclass>
    </joined-subclass>
    <subclass name="currentStudent"discrimintor-value="OSTS">
        <property name=" ostudents"/>
        <property name="  ocemail"/>
    <property name="  octc"type="double"/>
    </joined-subclass>
    </class>
</hibernate-mapping>
```

Jtc7A.java	Jtc7B.java
Student.java	currentStudent.java
oldStudent.java	Weekday student.java
WeekendStudent.java	ChibernateUtil.java
Student.hbm.xml	Hibernate.cfg.xml

Student.hbm.xml

```
<hibernate-mapping =”com.jtcindia.hibernate”>
<class name=”student”table=”jtcstudents”
Discriminator-value=”STU”>
<id name=”sid”column=’sid”type=”int”>
<generator colass=”increment”/>
</id>

<discriminator column=”stutype”/>
<property name=”sname”/>
<property name=”city”/>
<property name=”status”/>
<property name=”totalfee”type=”double”/>

<subclass name=”currentStudent”discriminator
value=”CSTU”>
<property name=”feebal”type=”double”/>
<property name=”timings”/>
<property name=”branch”/>

<subclass name=”regularStudent”discriminator
value=”RSTU”>
<property name=”qualification”/>
<property name=”percentage”/>
<property name=”yoe”type=”int”/>
</subclass>
<subclass name=”weekendStudent”discriminator
value=” WSTU”>
<property name=”wcompany”/>
<property name=”wcemall”/>
<property name=”wctc”type=”double”/>
</subclass>
</subclass>

<subclass name=”oldStudent”discriminator
value=”OSTU”>
<property name=”ocompany”/>
<property name=”octc”type=”double”/>
</subclass>
</class>
</hibernate-mapping>
```

Example with hibernate annotation (Jtc8)

B) Hibernate persistence classes

1. Student.java

```
@Entity
@ Table (name=”jtcstudents”)
@Inheritance (strategy=Inheritance Type. SINGLE_JOINED)
@Discriminator column (name = “stu Type”, length = 5)
@Discriminator value (value = “STU”)
Public class student {
…..
}
```

2. Current student.java

```
@Entity
@ Table (name=”customers”)
@Discriminator value (value = “CSTU”)
Public class current student extends student {
……
}
```

3 .Old student.java

```
@Entity
@ Table (name=”ostudents”)
@Discriminator value (value = “OSTU”)
Public class old student extends student {
……
}
```

4 .Weekday student.java

```
@Entity
@ Table (name=”wdstudents”)
@Discriminator value (value = “WDSTU”)
Public class weekday student extends student {
……
}
```

4 .Weekday student.java

```
@Entity
@ Table (name=”wdstudents”)
@Discriminator value (value = “WESTU”)
Public class weekday student extends student {
……
}
```

Note:

- In the case of Table per sub class mapping,
  - Use the following for super class
    @ Inheritance (strategy = Inheritance Type. SINGLE_TABLE)
    @discriminator column (name = “stu Tope”, length = 5)
    @Discriminator value (value = “xx”)
  - Use the following for super class and all the sub classes
    @Discriminator value (value = “xx”)

Jtc8: Files required

Jtc8a.java	Jtc8B.java
Student. java	Current student.java
Old student. java	Weekday student.java
Weekend Student.java	AHibernate Util.java
Hibernate. Cfg. xml	

Student.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”jtcstudents”)
@Inheritance(strategy=InheritanceType.SINGLE_TABLE)
@DiscriminatorColumn(name=”stuType”,length=5)
@DiscriminatorValue(value=”STU”)
Public class student{
@id
@Column(name=”sid”)
@GeneratedValue(strategy=generationType.AUTO)
Private int sid;
@column(name=”sname”)
Private string sname;
@Column(name=”city”)
Private string city
@Column(name=”status”)
Private string status;
@Column(name=”totalfee”)
Private double totalfee;

Public student() {}

Public student (string sname, string city, string status,
double totalfee) {
This.sname=sname;
This.city=city;
This status = status;
This. Totalfee=totalfee;
}

//setter and getter
@override
```

CurrentStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@DiscriminatorValue(value=”CSTU”)
Public class currentStudent extends
student{
@column(name=”feebal”)
Private double feebal;
@column(name=”timings”)
Private string timings;
@private string branch”)
Private string branch;

Public currentStudent(){
}

Public currentStudent (string sname,string
city, string status,
Double totalfee, double feebal, string
timings, string branch){
Super(sname, city, status, totalfee);
This. Feebal=feebal;
This.timings=timings;
This.branch=branch;
}

//setter and getter
@override
Public string to string(){
+”,”+feebal+”,”+timings+”,”+branch;
}
}
```

```
Public string to string(){
Return sid+”,”+sname+”,”+city+”,”+status=”,”+totalfee;
}
}
```

OldStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@DiscriminatorValue(value=”OSTU”)
Public class OldStudent extends Student{
Private string company;
Private string email;
Private string ctc;

Public OldStudent () {}

Public weekend Student (string sname, string city,
string status,
Double totalfee, double feebal, string timings, strig
branch,
String company, string email, string ctc){
Super(sname,city, status, totalfee, feebal, timings,
branch);
This.company=company;
This.email=email;
This.ctc=ctc;
}

//Setter and getter

@Override
Public string to string(){
Return super.to
string()+”,”+company+”,”+email+”,”ctc;
}
}
```

WeekdayStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@DiscriminatorValue(value=”WDSTU”)
Public class weekdayStudent extends
CurrentStudent {
Private string qualificatio;
Private string percentage;
Private string yop;

Public WeekdayStudent () {}

Public weekend Student (string sname, string city,
string status,
Double totalfee, double feebal, string timings, strig
branch,
String company, string email, string ctc){
Super(sname,city, status, totalfee, feebal, timings,
branch);
This. qualificatio = qualificatio;
This. percentage = percentage;
This. yop = yop;
}

//Setter and getter

@Override
Public string to string(){
Return super.to string()+”,”+qualificatio +”,”+
percentage +”,” yop;
}
```

WeekendStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@DiscriminatorValue(value=”WESTU”)
Public class OldStudent extends Student{
Private string company;
Private string email;
Private string ctc;

Public OldStudent () {}

Public weekend Student (string sname, string city, string status,
Double totalfee, double feebal, string timings, strig branch,
String company, string email, string ctc){
Super(sname,city, status, totalfee, feebal, timings, branch);
This.company=company;
This.email=email;
This.ctc=ctc;
}

//Setter and getter

@Override
Public string to string(){
Return super.to string()+”,”+company+”,”+email+”,”ctc;
}
}
```

3. Table per concrete class mapping

- In this mapping, you need to take one table for one concrete class.
- When you save student class object then only one record will be inserted in mystudents1
- When you save current student class object then only one record will be inserted in cstudents 1
- When you save OLD student class object then only one record will be inserted in ostudents 1.
- When you save weekday student class object then only one record will be inserted in wdstudents 1.
- When you save weekend student class object then only one record will be inserted in westudents 1.

A) Tables required

1) My students

Sid	Sname	City	Status	Totalfee

2) Cstudents

Sid	Sname	Sity	Status	Totalfee	FEEBAL	TIMINGS	BRANCH

3) Ostudents

Sid	Sname	Sity	Status	Totalfee	Oscompany	Osemail	Osctc

4) Wdstudents

Sid	Sname	City	Status	Totalfee	Feebal	Timings	Branch	Qualification	Percentage

5) Westudents

Sid	Sname	City	Status	Totalfee	Feebal	Timings	Branch	Wecompany	Weemail

Example with Hibernate core (Jtc9)
B) Hibernate persistence classes
1. Public class student {
Private Int Sid;
Private string sname;
Private string city;
Private string status;
Private double totalfee;
…..
}
2. Current student.java
Public class current student extends student {
Private double feebal;
Private string timings;
Private string branch;
}
3. Old student.java
Public class Old student extends student {
Private string ocompany;
Private string ocemail;
Private double octc;

}
Weekday student.java
Public class Weekday student extends Current student {
Private string qualification;
Private string percentage;
Private Int yoe;
}
Weekend student.java
Public class Weekday student extends Current student {
Private string wcompany;
Private string wcemail;
Private double wctc;

Mapping Document

```
<hibernate-mapping package=”com.jtcindia.hibernate”>
        <class name=”Student”table=”mystudents”>
                <id name=”sid”column=”sid”type=”int”>
                        <generator coass=”increment”/>
                </id>
            <discriminator column=”stutype”/>
          <property name=”city”/>
         <property name=”status”/>
        <property name=”totalfee”type=double”/>
</class>
        <class name=”currentStudent”table=”cstudents1”>
            <id name=”sid”column=”sid”type=”int”>
                <generator class=”increment”/>
              </id>
              <property name=”sname”/>
              <property name=”city”/>
              <property name=”status”/>
              <property name=”totalfee”type=double”/>
              <property name=”feebal”type=”double”/>
              <property name=”timings”/>
              <property name=”branch”/>
</class>
        <class name=”OldStudent”table=”ostudents1”>
            <id name=”sid”column=”sid”type=”int”>
                <generator class=”increment”/>
              </id>
              <property name=”sname”/>
              <property name=”city”/>
              <property name=”status”/>
              <property name=”totalfee”type=double”/>
              <property name=”ocompany”/>
              <property name=”ocemail”/>
              <property name=”octc”type=”double”/>
</class>
        <class name=”regularStudent”table=”rstudents1”>
            <id name=”sid”column=”sid”type=”int”>
                <generator class=”increment”/>
              </id>
              <property name=”sname”/>
              <property name=”city”/>
              <property name=”status”/>
              <property name=”totalfee”type=double”/>
              <property name=”feebal”type=”double”/>
              <property name=”timings”/>
              <property name=” branch”/>
              <property name=”qualification”/>
              <property name=”percentage”/>
              <property name=”yoe”type=”int”/>
```

```
</class>
        <class name=”weekedStudent”table=”wstudents1”>
            <id name=”sid”column=”sid”type=”int”>
                <generator class=”increment”/>
             </id>
             <property name=”sname”/>
             <property name=”city”/>
             <property name=”status”/>
             <property name=”totalfee”type=double”/>
             <property name=”feebal”type=”double”/>
             <property name=”timings”/>
             <property name=” branch”/>
             <property name=”wcompany”/>
             <property name=”wcemail”/>
             <property name=”yoe”type=”double”/>
```

Jtc9: Files required

Jtc5a.java	Jtc5B.java
Student. java	Current student.java
Old student. java	Weekday student.java
Weekend Student.java	Hibernate Util.java
Student. Hbm. xml	Hibernate. Cfg. xml

```
<hibernate-mapping
package=”com.jtcindia.hibernate”>
  <class name=”Student”table=”mystudents”>
  <id name=”sid”column=”sid”type=”int”>
  <generator coass=”increment”/>
  </id>
  <discriminator column=”stutype”/>
  <property name=”city”/>
  <property name=”status”/>
  <property name=”totalfee”type=double”/>
  </class>
```

```
</class>
<class name=”weekedStudent”table=”wstudents1”>
<id name=”sid”column=”sid”type=”int”>
<generator class=”increment”/>
</id>
<property name=”sname”/>
<property name=”city”/>
<property name=”status”/>
<property name=”totalfee”type=double”/>
<property name=”feebal”type=”double”/>
<property name=”timings”/>
<property name=” branch”/>
<property name=”wcompany”/>
<property name=”wcemail”/>
<property name=”yoe”type=”double”/>
```

```
<class name=”currentStudent”table=”cstudents1”>
<id name=”sid”column=”sid”type=”int”>
<generator class=”increment”/>
</id>
<property name=”sname”/>
<property name=”city”/>
<property name=”status”/>
<property name=”totalfee”type=double”/>
<property name=”feebal”type=”double”/>
<property name=”timings”/>
<property name=”branch”/>
<class name=”OldStudent”table=”ostudents1”>
<id name=”sid”column=”sid”type=”int”>
<generator class=”increment”/>
</id>
<property name=”sname”/>
<property name=”city”/>
<property name=”status”/>
<property name=”totalfee”type=double”/>
<property name=”ocompany”/>
<property name=”ocemail”/>
<property name=”octc”type=”double”/>
</class>
<class name=”regularStudent”table=”rstudents1”>
<id name=”sid”column=”sid”type=”int”>
<generator class=”increment”/>
</id>
<property name=”sname”/>
<property name=”city”/>
<property name=”status”/>
<property name=”totalfee”type=double”/>
<property name=”feebal”type=”double”/>
<property name=”timings”/>
<property name=” branch”/>
<property name=”qualification”/>
<property name=”percentage”/>
<property name=”yoe”type=”int”/>
```

Example with hibernate annotation (Jtc10)

B) Hibernate persistence classes

1. Student.java

```
@Entity
@ Table (name=”mystudents”)
@Inheritance (strategy=Inheritance Type. TABLE_PER_CLASS)
Public class student {
@ Column (name = “sid”)
Private Int Sid;
…..
}
```

2. Current student.java

```
@Entity
@ Table (name=”customers”)
Public class current student extends student {
……
}
```

3 .Old student.java

```
@Entity
@ Table (name=”ostudents”)
Public class old student extends student {
……
}
```

4 .Weekday student.java

```
@Entity
@ Table (name=”wdstudents”)
Public class weekday student extends student {
……
}
```

5 .Weekday student.java

```
@Entity
@ Table (name=”westudents”)
Public class weekday student extends student {
……
```

}

Note:

- In the case of Table per sub class mapping,
  - Use the following for super class
    @Entity
    @Table
    @ Inheritance (strategy = Inheritance Type. TABLE_PER_CLASS)

  - Use the following for super class
    @Entity
    @Table
  - Don't specify the primary key generation for Sid.
  - You set the Sid value to all the objects in the client code.
  - Don't save all types of student class objects in one session.

Jtc10: Files required

Jtc10a.java	Jtc10B.java
Jtc10c.java	Jtc10d.java
Jtc10e.java	Jtc10f.java
Student. java	Current student.java
Old student. java	Weekday student.java
Weekend Student.java	Hibernate Util.java
Hibernate. Cfg. xml	

Jtc10A.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
Public class lab10A{
Public static void main(string[]args){
Transaction tx=null;
Try{
SessionFactory
sf=AhibernateUtil.getSessionFactory();
Session session=sf.openSession();
Tx=session.begintransaction();
Student st=new
student("som","noida","Active","14000);
Session.sava(st);
Tx.commit();
Session.close();
```

Jtc10B.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
Public class lab10A{
Public static void main(string[]args){
Transaction tx=null;
Try{
SessionFactory
sf=AhibernateUtil.getSessionFactory();
Session session=sf.openSession();
Tx=session.begintransaction();
CurrentStudent cs =new
CurrentStudent("Rai"."noida","
Active",17000,5000,7:00-9:00","Sec-2");
Cs.setSid(1);
Session.close();
```

```
}catch(exception e) {
e.printStackTrace();
tx.rollback();
}
}
}
```

```
}catch(exception e) {
e.printStackTrace();
tx.rollback();
}
}
}
```

Jtc10C.java

```
Package com.jtcindia.hibernate;
Import java.util.*;
/*
*@Author:Som Prakash Rai
*@company:java Training Center
*@See          : www.jtcindia.org
**/
Public class Jtc10C{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session Factory();
session session=sf.openSession();
tx=session.beginTransaction();

oldStudent os=new
oldStudent("Rai1","noida","Active",15000,"JTC","Prakash
@jtc.org","5.6L");

Os.setSid(1);

Session.sava(os);
Tx.commit();
}catch(exception e) {
e.prinStackTrace();
tx.rollback();
}
}
}
```

Jtc10D.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training Center
*@See          : www.jtcindia.org
**/
Public class Jtc10D{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session
Factory();
session session=sf.openSession();
tx=session.beginTransaction();

Weekday student wds=new weekday
student('SP","Noida","active",14000,"10:00-
2:00","NOIDA","Mtech","98%","1999");

Os.setSid(1);

Session.sava(os);
Tx.commit();
}catch(exception e) {
e.prinStackTrace();
tx.rollback();
}
}
}
```

Jtc10E.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training Center
*@See        : www.jtcindia.org
**/
Public class Jtc10D{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session Factory();
session session=sf.openSession();
tx=session.beginTransaction();

Weekday student wds=new
Weekday student(student",”Noida”,” active”, 20000,5000”
8:00-2:00”,”Sec-2”,”12.5L”);

Os.setSid(1);

Session.sava(os);
Tx.commit();
}catch(exception e) {
e.prinStackTrace();
tx.rollback();
}
}
}
```

Jtc10F.java

```
Package com.jtcindia.hibernate;
Import org.hibernate.*;
/*
*@Author:Som Prakash Rai
*@company:java Training Center
*@See        : www.jtcindia.org
**/
Public class Jtc10D{
Public static vaid main(string{} args){
Transation tx=null;
Try{
sessionFactory sf=Ahibernate Util.get Session
Factory();
session session=sf.openSession();
tx=session.beginTransaction();

Student st=session.load(weekendStudent.class,1);
System.out.printin(st);

Session.sava(os);
Tx.commit();
}catch(exception e) {
e.prinStackTrace();
tx.rollback();
}
}
}
```

Student.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”my students1”)
@Inheritance(strategy=InheritanceType.JOINED)
Public class student{
@id
@Column(name=”sid”)
@GeneratedValue(strategy=generationType.AUTO)
Private int sid;
@column(name=”sname”)
Private string sname;
@Column(name=”city”)
Private string city
@Column(name=”status”)
Private string status;
@Column(name=”totalfee”)
Private double totalfee;

Public student() {
}

Public student (string sname, string city, string status,
double totalfee) {
This.sname=sname;
This.city=city;
This status = status;
This. Totalfee=totalfee;
}

//setter and getter
@override
Public string to string(){
Return sid+”,”+sname+”,”+city+”,”+status=”,”+totalfee;
}
}
```

CurrentStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”cstudents1”)
@primaryKeyJoinColumn(name=”sid”)
Public class currentStudent extends student{
@column(name=”feebal”)
Private double feebal;
@column(name=”timings”)
Private string timings;
@private string branch;
Private string branch;

Public currentStudent(){
}

Public currentStudent (string sname,string city, string
status,
Double totalfee, double feebal, string timings, string
branch){
Super(sname, city, status, totalfee);
This. Feebal=feebal;
This.timings=timings;
This.branch=branch;
}

//setter and getter
@override
Public string to string(){
+”,”+feebal+”,”+timings+”,”+branch;
}
}
```

oldStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”ostudents1”)
Public class weekendStudent extends currentStudent{
Private string company;
Private string email;
Private string ctc;

Public weekendStudent() {}

Public weekend Student (string sname, string city, string
status,
Double totalfee, double feebal, string timings, strig ctc,
String company, string email, string ctc){
Super(sname,city, status, totalfee);
This.company=company;
This.email=email;
This.ctc=ctc;
}

//Setter and getter

@Override
Public string to string(){
Return super.to
string()+”,”+company+”,”+email+email+”,”ctc;
}
}
```

WeekendStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”westudents”)
Public class weekendStudent extends currentStudent{
Private string quallfication;
Private string percentage;
Private string yop;

Public weekendStudent() {}

Public weekend Student (string sname, string city,
string status,
Double totalfee, double feebal, string timings, strig
branch,
String company, string email, string ctc){
Super(sname,city, status, totalfee, feebal, timings,
branch);
This. quallfication = quallfication;
This. percentage = percentage;
This. yop = yop;
}

//Setter and getter

@Override
Public string to string(){
Return super.to string()+”,”+quallfication +”,”+
percentage +”,” yop;
}
}
```

WeekendStudent.java

```
Package com.jtcindia.hibernate;
Import javax.persistence.*;
/*
*@Author:Som Prakash Rai
*@Company:Java Training Center
*@See          :www.jtcindia.org
**/
@Entity
@Table(name=”westudents1”)
Public class weekendStudent extends currentStudent{
Private string company;
Private string email;
Private string ctc;

Public weekendStudent() {}

Public weekend Student (string sname, string city, string status,
Double totalfee, double feebal, string timings, strig branch,
String company, string email, string ctc){
Super(sname,city, status, totalfee, feebal, timings, branch);
This.company=company;
This.email=email;
This.ctc=ctc;
}

//Setter and getter

@Override
Public string to string(){
Return super.to string()+”,”+company+”,”+email+”,”ctc;
}
}
```