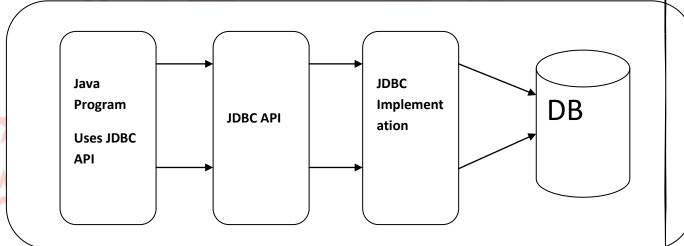
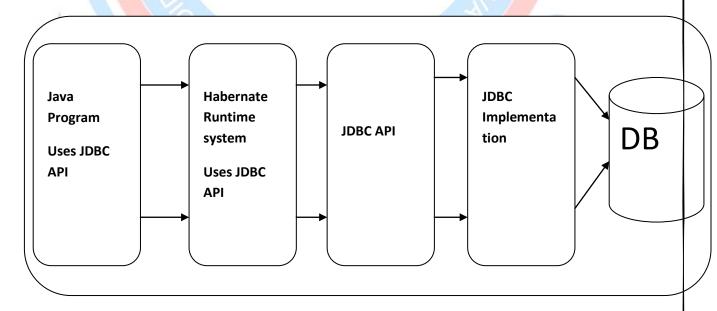
(No 1 in Training & Placement)

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.



(No 1 in Training & Placement)

• 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.

(No 1 in Training & Placement)

- 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. OBE
 - d. Native SQL
 - e. Named SQL

better performance.

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

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 Jucindia db; USE Jtcindia db;

CREATE TABLE customers (

(No 1 in Training & Placement)

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 ON, AC7/0	Customer.java
Customer, hbm.xml	Hibernate. cgf.xml	CHibernate Util. java

JTC1:	2.Jtc1B.java
	==== spara ===================================
package com.jtc <mark>icin</mark> dia.hibernate;	package com.jtcicindia.hibernate;
import org.hibernate.*;	import org.hibernate.*;
/*	/*
*@Author:Som Prakash Rai	*@Author:Som Prakash Rai
*@company:java Training center	*@company:java Training center
*@See :Www.jtcindia.org	*@See :Www.jtcindia.org
*@call :9990399111	*@call :9990399111
**/	\/\\ \\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
public class Jtc1A{	public class Jtc2B{
<pre>public static void main(string[]args){</pre>	public static void main(string[]args){
Transaction tx=null;	Transaction tx=null;
try{	try{
SessionFactory	SessionFactory
sf=CHibernateUtil.getSessionFactory();	sf=CHibernateUtil.getSessionFactory();
Session session=sf.openSession();	Session session=sf.openSession();
tx=session.beginTransaction();	tx=session.beginTransaction();
Customer cust=new	
customer("som",som@jtc",123,"Noida",25000);	Customer cust =(Customer)
session.save(cust);	session.load(Customer.class,1);
tx.commit();	system.out.println(cust.getCld()+"/t"+cust.getCname()
session.close();	+"\t"+cust.getEmail()+cust.geatphone());
System.out.println("record inserted");	
)catch(Exception e){	tx.commit();
e.prinStackTrace();	}catch(Exception e){
<pre>if(txl=null)tx.rollback();</pre>	e.printStackTrace();
}	<pre>if(tx!=null)tx.rollback();</pre>
}}	}
<u>Jan 2000</u>	

```
3.Customer.java
                                                5.hibernate.cfg.xml
package com.jtcindia.hibernate;
public class Customer{
                                                <hibernate-configuration>
Private int cid;
                    //P.K
                                                <session-factory>
private String cname;
                                                cproperty name="hibernate.connection.driver_class">
                                                        com.mysql.jdbc.Driver
private string email;
private long phone;
                                                 </property>
private string city;
                                                 propety name="hibernate.connection.url">
                                                        jdbc:mysql://localhost:3306/jtcindiadb
private double bal;
                                                 </property>
public Customer() {}
                                                 property name="hibernate.connection.username">
Public customer(string cname, string email, long
phone,
                                                 </property>
String city, double bal)
                                                property name="hibernate.connection.password">
this.cname=cname;
                                                        somsree
this.email=email;
                                                 </property>
this.phone=phone;
                                                 cproperty name="dialect">
                                                        org.hibernate.dialect.MySQLDialect
this.city=city;
this.bal=bal;
                                                </property>
                                                cproperty name="show_sql">true</property>
       //Setters and getter
                                                 <mapping
                                                resource="com/jtcindia/hibernate/customer.hbm.xml/>
4.customer.hbm.xml
                                                </session-factory>
                                                </hibernate-configuration>
<hibernate-mapping
                                                6.cHibernateUtil.java
package="com.itcindia.hibernate">
<class name="customer" table="customer">
                                                package com.jtcindia.hibernate;
<id name="cid"column="cid"type="int">
                                                import org.hibernate.*;
<generator class="increment"/>
</id>
                                                *@Author:Som Prakash Rai
cproperty name="cname"/>
                                                 *@company:java Training center
cproperty name="email"column="email"/>
                                                 *@See :Www.jtcindia.org
                                                 *@call:9990399111
property
                                                 **/
name="phone"cloumn=phone"type="long"/>
                                                public class CHibernateUtil{
name="city"column="city"type="string"/>
                                                static SessionFactory factory;
                                                static{
property
name="bal"column="bal"type="double"/>
                                                try{
                                                Configuration cfg=new Configuration();
</class>
                                                cfg=cfg.configure();
</hibernate-mapping>
                                                factory=cfg.buildSessionFactory();
                                                 }catch(Exception e){e.printStacktrace();}
                                                Public static SessionFactory getSessionFactory(){
                                                return factory; } }
```

(No 1 in Training & Placement)

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. <u>Note:</u> 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 Itcindia 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 K 1D 1	2 T. 1D:
1.Jtc1B.java	2.Jtc1B.java
package com.jtcicindia.hibernate; import org.hibernate.*; /* *@Author:Som Prakash Rai *@company:java Training center	package com.jtcicindia.hibernate; import org.hibernate.*; /* *@ Author:Som Prakash Rai *@company:java Training center
*@See :Www.jtcindia.org *@call :9990399111 **/	*@See :Www.jtcindia.org *@call :9990399111 **/
<pre>public class Jtc2B{ public static void main(string[]args){ Transaction tx=null;</pre>	<pre>public class Jtc2B{ public static void main(string[]args){ Transaction tx=null;</pre>
try{ SessionFactory sf=AhibernateUtil.getSessionFactory(); //Same as Jtc1A.java	try{ SessionFactory sf=AhibernateUtil.getSessionFactory(); //Same as Jtc2A.java
<pre>}catch(Exception e){ e.printStackTrace(); if(tx!=null)tx.rollback();}}</pre>	<pre>}catch(Exception e){ e.printStackTrace(); if(tx!=null)tx.rollback();}}}</pre>
3.Customer.java Package com.jtcindia.hibernate; Import javax.persistence.*;	4.hibernate.cfg.xml ====================================

(No 1 in Training & Placement)

```
<session-facttory>
                                                    property
*@Author:Som Prakash Rai
*@company:java Training center
                                                   name="hibernate.connection.driver class">
*@See :Www.jtcindia.org
                                                          Com.mysql.jdbc.Driver
*@call:9990399111
                                                   **/
                                                    property name="hibernate.connection.url">
                                                           Jdbc:mysql://localhost:3306/jtcindiadb
@Entity
@table(name="customers")
                                                    </property>
Public class customer{
                                                   property name="hibernate.connection.username">
@id
                                                          Root
@GeneratedValue(strategy=GenerationType.AUTO)
                                                   </property>
                                                    property name="hibernate.connection.password">
@Column(name="cid")
Private int cid;//P.K
                                                      somsree
@Column(name="cname")
                                                   </property>
Private string cname;
                                                   property name="dialect">
                                                           Org.hibernate.dialect.MySQLDialect
@Column(name="email")
                                                    </property>
Private string cname;
                                                    <pcoperty name="show sql">true</property>
                                                   <mapping
                                                   class="com.jtcindia.hibernate.Customer"/>
(a)Column(name="phone")
Private string cname;
                                                   </session-factory>
                                                    </hibernate-configuration>
@Column(name="city")
Private string cname;
                                                   5.AhibernateUtil.java
@Column(name="bal")
                                                   Package com.jtcindia.hibernate;
Private string cname;
                                                   Import org.hibernate.*;
                                                   Import org.hibernate.cfg.*;
Public customer() {}
                                                   /*NBOUND
Public Customer(String cname, String email, long
                                                    *@author:Som Prakash Rai
phone, String city, double bal){
                                                    *@comapany:Java Training Center
this.cname=cname:
                                                   *@See
                                                                 :www.jtcindia.org
this.email=email;
                                                    **/
this.phone=phone;
                                                   Public class AhibernateUtil{
this city=city;
                                                   Static sessionFactory factory;
this.bal=bal;
                                                   Static{
                                                   AnnotationConfiguration cfg=new
     //Setters and Getters
                                                   annotationConfiguration();
                                                   cfg=(AnnotationConfiguration)cfg.configure();
                                                   factory=cfg.buildSessionFactory();
                                                   Public static SessionFactory setSessionFactory() {
                                                   Return factory;
```

Questions

(No 1 in Training & Placement)

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);
.......
GROWTH UNBOUND
```

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?

(No 1 in Training & Placement)

```
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:

(No 1 in Training & Placement)

```
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.

```
column="email"/>

column="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

(No 1 in Training & Placement)

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

Refs

C_{ℓ}	711	rs	20	

	74.4		
Sid	1	Cname	Idx
101		Java	0
101		JDBC	1
101		JSP	2

ľ	Sid	R name	R
	1/2		phone
ľ	No.	15	
١,	101	Aaa	1111
	Mary and	72	
	101	Bbb	2222
i	101	CCC	3333

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

TM

Marks

Sid	Marks
101	100
101	99
101	100

D) Client Code

String[]crs={"Java","JDBC","JSP"};

List<string>ems=new ArrayList<String>();

11

Hibernate Core/Annotation(P1) Author:Som Prakash Rai

www.jtcindia.org

```
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
       <a href="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"/>
cproperty name="dob"/>
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>
       <br/><br/>dag name="marks"<br/>table=table="marks">
              <key clolumn="sid"/>
              <element column="marks"type="int"/>
       <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"/>
                      </class> </hibernate-mapping>
       </map>
```

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

```
Jtc3A.java
                                                   Jtc3B.jav a
Package com.jtcindia.hibernate;
                                                   Package com.jtcindia.hibernate;
Import org.hibernate.*;
                                                   Import org.hibernate.*;
Import java.util.*;
                                                   Import java.util.*;
*@Author:Som Prakash Rai
                                                   *@Author:Som Prakash Rai
*@company:java Training center
                                                   *@company:java Training center
*@See :Www.jtcindia.org
                                                   *@See :Www.jtcindia.org
*@call:9990399111
                                                   *@call:9990399111
Public class Jtc3A {
                                                   Public class Jtc3A {
Public static void main (string[]args){
                                                   Public static void main (string[]args){
SessionFactory sf=ChibernateUtil.getFactory();
                                                   SessionFactory sf=ChibernateUtil.getFactory();
Session session=sf.openSession();
                                                   Session session=sf.openSession();
Transaction tx=session.beginTransaction();
                                                   Transaction tx=session.beginTransaction();
String cous[]={"java","JDBC","JSP"];
                                                   Student stu=(Student)session.load(student.class,1)
List<string>ems=new ArrayList<string>();
                                                   System.out.println(stu.getSid()+"\t"+stu.getSname()
Ems.add("aa@itc");
                                                   +"\" t+stu.getDob()){
Ems.add("bb@jtc");
                                                   Ststem.out.println(cn);
Ems.add("cc@jtc");
List<Integer>maks=new ArrayList<Integer>();
                                                   System.out.println(sty,get/enauks());
Maks.add(new Integer(100));
                                                   System.out.println(sty,get/Marks());
Maks.add(new Integer(99));
                                                   System.out.println(sty,get/phones());
Maks.add(new Integer(100));
                                                   System.out.println(sty,get/refs());
Set<Long>phs=new HashSet<long>();
                                                   Tx.commit():
Phs.add(new Long(1111));
                                                   Session.close():
Phs.add(new Long(2222));
                                                   }catch(Exception e){
Phs.add(new Long(3333));
                                                  e.printStackTracke();
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();
```

Java Training Center (No 1 in Training & Placement)

Student.java	Student.hbm.xml
Package com.jtcindia.hibernate;	 https://www.energia.com/<a com.jtcindia.hibernate"="" href="https://www.energ</td></tr><tr><td>Import java.util.*;</td><td>package=">
/*	<pre><class name="student" table="students"></class></pre>
*@Author:Som Prakash Rai	<id>id name="sid"column="sid"type="int"></id>
*@company:java Training Center	<pre><generator class="increment"></generator></pre>
*@See : www.jtcindia.org	
**/	<pre><pre><pre><pre><pre><pre>property name="sname"/></pre></pre></pre></pre></pre></pre>
Public class Student{	<pre><pre><pre><pre><pre><pre><pre>property name="dob"/></pre></pre></pre></pre></pre></pre></pre>
Private int sid;//P.K	<pre><pre><pre><pre><pre><pre><pre>property name="qualification"column="quali/></pre></pre></pre></pre></pre></pre></pre>
Private string sname;	<pre><array name="courses" table='courses"'></array></pre>
Privatge stringdob;	<pre><key column="sid"></key></pre>
Private string qualification;	<index column="idx"></index>
Private string[] courses;	<pre><element column="cname" emails"table="emails" type="string/></pre></td></tr><tr><td></td><td></td></tr><tr><td>Private list String > emails;</td><td></array></td></tr><tr><td>PrivateList<Integer>marks;</td><td><pre>t name="></element></pre>
Private set <long>phones;</long>	<pre><key column="sid"></key> cinday column="idy"/></pre>
Private map <string,long>refs;</string,long>	<pre><index column="idx"></index> </pre>
Public Student(){}	<pre><element column="marks" type="string"></element></pre>
Public Student(String sname, string dob,	
stringqualifaction,	<pre> dag name="marks"table="marks"> </pre>
String[]courses,	<key column="sid"></key>
List <string>emails,List<integer>Marks,</integer></string>	<pre><element column="marks" type="int"></element></pre>
Set < long > phones, map < string, Long > regs)(super();	
This.sname = sname;	<pre><set name="phones" table='phones".</pre'></set></pre>
This.dob =dob;	<key column="sid"></key>
This.qualification=qualification;	<pre><element column="phoneNo" type="long"></element></pre>
This.courses = courses;	
This.emails =marks; GROW	<map name="" refs"="" table="refs"></map>
This.phones=phones;	<pre><key column="rphone" type="long"></key></pre>
This.refs = refs;	
}	
//setters and getters	
} Hibernate.cfg.xml	W. W.
	MAD ENIN
<hibernate-configuration></hibernate-configuration>	NEO O.
<session-factory></session-factory>	
<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	
name="hbm2ddl.auto">update	
<maping< td=""><td></td></maping<>	
resource="com/jtcindia/hibernate/student.hbm.xm"	
1"/>	

(No 1 in Training & Placement)

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

(No 1 in Training & Placement)

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

GROWTH LINBOLING

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

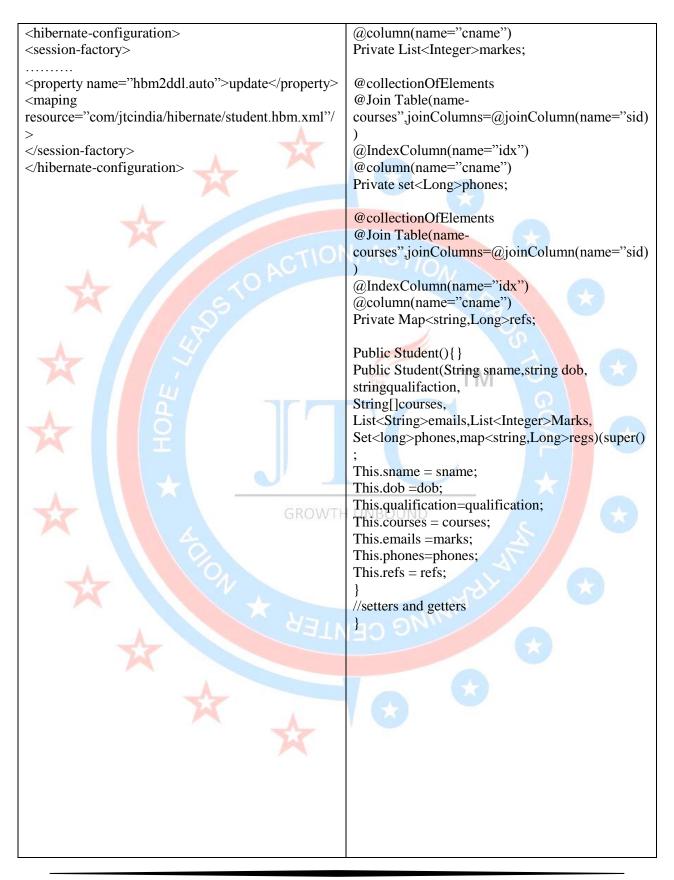
Jtc4:Files required

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

16

Java Training Center (No 1 in Training & Placement)

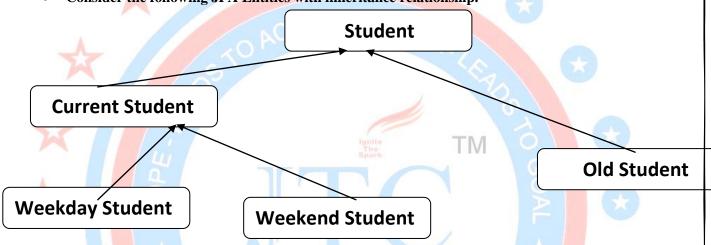
Jtc4A.java	Student.java
Package com.jtcindia.hibernate;	Package com.jtcindia.hibernate;
Import java.util.*;	Import javax.persistence.*;
/*	Import java.util.*;
@Author:Som Prakash Rai	/
*@company:java Training center	*@Author:Som Prakash Rai
*@See :Www.jtcindia.org	*@company:java Training center
*@call :9990399111	*@See :Www.jtcindia.org
**/	*@call:9990399111
Public class Jtc4A{	**/
Public static vaid main(string{} args){	@Entity
Transation tx=null;	@Table(name="MyStudents")
Try{	Public class Student(
sessionFactory sf=Ahibernate Util.get Session	@Id
Factory();	@Generated
//Same as Jtc3A.java	Value(strategy=GenerationType.AUTO)
}catch(exception e){	@Column(name="sid")
e.printStack Trace();	Private int sid;//P.K
if(tx!=null)tx.rollback();	
}	@ColumnI(name="sname")
	Private string sname;
Jtc4B.java	Spark
Package com.jtcindia.hibernate;	@column(name="dob")
Import java.util.*;	Private string dob;
/* /*	
*@Author:Som Prakash Rai	@column(name="quali")
*@company:java Training center	Private string qualification;
*@See :Www.jtcindia.org	
*@call :9990399111 GROWTH	@collectionOfElements
**/	@Join Table(name-
Public class Jtc3B{	courses",joinColumns=@joinColumn(name="sid)
Public static void main(string[]args){	
Transaction tx=null;	@IndexColumn(name="idx")
Try{	@column(name="cname")
sessionFactory	Private string[]courses;
sf=AhibernateUtill.gettSessionFactory();	ING CE
//Same as Jtc3B.java	@collectionOfElements
}cetch(Exception e){	@Join Table(name-
e.prinStack Tarace();	courses",joinColumns=@joinColumn(name="sid)
if(tx!=null)tx.rollback();	
}}}	@IndexColumn(name="idx")
***	@column(name="cname")
X	Private List <string>emails;</string>
	@collectionOfElements
	@Join Table(name-
	courses",joinColumns=@joinColumn(name="sid)
Hibernate.cfg.xml	(a) (a) (a) (a) (b) (d) (d) (d) (d) (d) (d) (d) (d) (d) (d



(No 1 in Training & Placement)

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.
 - o One record will be inserted in my students.
 - One record will be inserted in students
- When you save Old Student class object then.
 - o One record will be inserted in my students.
 - o One record will be inserted in ostudents.
- When you save weekday student class object then
 - o One record will be inserted in my students.
 - One record will be inserted in csteudents.
 - o One record will be inserted in wdstudents.

(No 1 in Training & Placement)

- O 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
70				

2) Csteudents

Sid	Feebal	Timings	Branch
N A			*
	CTION, A	CTI	

3) Ostudents

Sid	Oscompany	Osemail	Osctc
		2	

4) Wdstudents

Sid	Qualification	Percentage	yop
		T-0.4	

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;

_..A.

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

20

Hibernate Core/Annotation(P1) Author:Som Prakash Rai

www.jtcindia.org

(No 1 in Training & Placement)

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

stes. The 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



```
//5.Adding the WeekendStudent
Jtc5A.java
Package com.jtcindia.hibernate;
                                                 WeekendStudent wstu=new
Import java.util.*;
                                                 weekendStudent("cc","Noida","Enabled",15000.0,
/*
                                                 2000.0.
                                                 6.30P.M", "GRNOIDA" < "WEBEARN", cc@sp.com
*@Author:Som Prakash Rai
*@company:java Training center
                                                 .9.0);
*@See :Www.jtcindia.org
                                                 it =(integer)session.save(wstu);
*@call:9990399111
                                                 System.out.println(sty,get/refs());
**/
                                                 Tx.commit();
Public class Jtc5A{
                                                 Session.close();
Public static vaid main(string{} args){
                                                 }catch(Exception e){
Transation tx=null;
                                                 e.printStackTracke();
                                                 if(tx!=null)tx.rollback();
Try{
                                                 3N, AC
sessionFactory sf=Ahibernate Util.get Session
getFactory();
Session session=factory.openSession();
                                                 }
Tx=session.beginTransaction();
                                                 Jtc5B.java
                                                 Package com.jtcindia.hibernate;
//1.Adding the student
Student stuenew
                                                 Import org.hibernate.*;
                                                 Import java.util.*;
Stuednt("som","Noida","Enabled",15000.0);
Integer it=(integer)session.save(stu);
System.out.println(it.intValue());
                                                 *@Author:Som Prakash Rai
                                                 *@company:java Training center
//2.Adding the CurrenStudent
                                                 *@See :Www.jtcindia.org
Current Student cstu=new
                                                 *@call:9990399111
CurrentStudent("Rai","Noida","Enabled",15000
.0, 2000.0, "6.30P.M", "Sec-2");
                                                 Public class Jtc3A {
It=(integer)session.save(cstu);
                                                 Public static void main (string[]args){
System.out.println(it.intValue());
                                                 SessionFactory sf=ChibernateUtil.getFactory();
                                                 Session session=sf.openSession();
//3.Adding the oldStudent
OldStudent ostu=new
                                                 tx=session.beginTransaction();
OldStudent
                                                 System.(string)session.load(student.class.1);
("aa","Noida","Enabled",15000.0,"WEBEARN
                                                 System.out.println(st);
", "aa@sp.com9.0);
                                                 Tx.commit();
It=(integer)session.save(ostu);
                                                 Session.close();
System.out.printin(it.intValue());
                                                 }catch(Exception e){
                                                 e.printStackTracke();
//4.Adding the regular student
                                                 if(tx!=null)tx.rollback();
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());
```

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

(No 1 in Training & Placement)

OldStudent.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;

WeekdayStudent.java

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;



Java Training Center (No 1 in Training & Placement)

WeekendStudent.java	Student.hbm.xml
Package com.jtcindia.hibernate;	<hibernate-mapping< td=""></hibernate-mapping<>
Import java.util.*;	package="com.jtcindia.hibernate">
/*	<pre><class name="student" table="mystudents"></class></pre>
*@Author:Som Prakash Rai	<id column="sid" name="sid" type="int"></id>
*@company:java Training center	<pre><generator class="Increment"></generator></pre>
*@See :Www.jtcindia.org	
*@call :9990399111	<pre><pre><pre><pre><pre><pre>property name="sname"/></pre></pre></pre></pre></pre></pre>
**/	<pre><pre><pre><pre><pre><pre><pre>city"/></pre></pre></pre></pre></pre></pre></pre>
Public class weekend Student extends	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
currentStudent{	<pre><pre><pre><pre>cycle in the company in the com</pre></pre></pre></pre>
Private string wcompany;	<pre><joined-subclass <="" name="currentStudent" pre=""></joined-subclass></pre>
Private string wcemail;	Table"cstudents">
Private double wctc;	<key column="sid"></key>
Public weekendstudent(){}	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
Public weekendString sname, string city, string	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
status, double totalfee, double feebal, string timings,	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
string branch, string wcompany, string wcemail,	<pre><joined-subclass <="" name="weekdayStudent" pre=""></joined-subclass></pre>
double wctc){	Table="wdstudents">
Super(sname, city, status, totalfee, feebal, timings,	<key column="sid"></key>
branch);	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
This wcompany=wcompany;	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
This.wcompany, This.wcemail=wcemail;	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
This.weeman-weeman, This.wetc=wetc;	
Tills.wetc=wetc,	<pre><joined-subclass <="" name="weekendStudent" pre=""></joined-subclass></pre>
//setters and getterrs	Table="westudents">
@override	<pre><key column="sid"></key></pre>
	<pre><rp><key column="sid"></key></rp></pre> <pre>cyroperty neme="wcompany"/></pre>
Public string to string() { Return super. To	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
-4 () 122 22 1 11 122 22 1 - 4	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
string()+, +company+email+, +etc; GROWTH	<pre></pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <pre> </pre> <pre> <!--</td--></pre></pre></pre></pre></pre>
	<pre><joined-subclass <="" name="oldStudent" pre=""></joined-subclass></pre>
1	Table="ostudents">
Hibernate.cfg.xml	
	<pre><key column="ocompany"></key> <pre></pre></pre>
<hibernate-configuration></hibernate-configuration>	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
<session-factory></session-factory>	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>
i	<pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre>
<mapping< td=""><td></td></mapping<>	
resource="com/jtcindia/hibernate/student.hbm.xml"/>	

TM

(No 1 in Training & Placement)

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 {WTH UNBOUND
......
}
```

4 .Weekday student.java

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

Note:

(No 1 in Training & Placement)

- o In the case of Table per sub class mapping,
 - o Use the following for super class
 - @ Inheritance (strategy = Inheritance Type. JOINED)
 - o 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	1/1/4

Jtc6A.java	Jtc6B.java		
Package com.jtcindia.hibernate;	Package com.jtcindia.hibernate;		
Import java.util.*;	Import java.util.*;		
/*	/*		
*@Author:Som Prakash Rai	*@Author:Som Prakash Rai		
*@company:java Training Center	*@company:java Training Center		
*@See : www.jtcindia.org	*@See : www.jtcindia.org		
**/	**/		
Public class Jtc15A{	Public class Jtc15B{		
Public static vaid main(string{} args){	Public static vaid main(string{} args){		
Transation tx=null; GROWT	Transation tx=null;		
Try{	Try{		
sessionFactory sf=Ahibernate Util.get Session	sessionFactory sf=Ahibernate Util.get Session		
Factory();	Factory();		
// Same as Jtc5A	// Same as Jtc5A		
7 27.	ONIN'		
}catch(Exception e){	}catch(Exception e){		
e.printStackTrace();	e.printStackTrace();		
tx.rollback();	tx.rollback();		
}	}		
	}		

```
Student.java
                                                    CurrentStudent.java
Package com.jtcindia.hibernate;
                                                    Package com.itcindia.hibernate;
Import javax.persistence.*;
                                                    Import javax.persistence.*;
                                                    *@Author:Som Prakash Rai
*@Author:Som Prakash Rai
*@Company:Java Training Center
                                                     *@Company:Java Training Center
*@See
             :www.jtcindia.org
                                                     *@See
                                                                  :www.jtcindia.org
**/
                                                     **/
@Entity
                                                     @Entity
@Table(name="my students")
                                                    @Table(name="cstudents")
@Inheritance(strategy=InheritanceType.JOINED)
                                                    @primaryKeyJoinColumn(name="sid")
                                                    Public class currentStudent extends student{
Public class student{
@id
                                                    @column(name="feebal")
@Column(name="sid")
                                                    Private double feebal;
@GeneratedValue(strategy=generationType.AUTO)
                                                    @column(name="timings")
Private int sid:
                                                    Private string timings;
@column(name="sname")
                                                     @private string branch;
Private string sname;
                                                    Private string branch;
@Column(name="city")
Private string city
                                                    Public currentStudent(){
@Column(name="status")
Private string status;
@Column(name="totalfee")
                                                    Public currentStudent (string sname, string city,
Private double totalfee;
                                                    string status,
                                                    Double totalfee, double feebal, string timings,
Public student() {
                                                    string branch){
                                                    Super(sname, city, status, totalfee);
                                                    This. Feebal=feebal;
Public student (string sname, string city, string
                                                    This.timings=timings;
status, double totalfee) {
                                                    This.branch=branch;
This.sname=sname;
This.city=city;
This status = status;
                                                    //setter and getter
This. Totalfee=totalfee;
                                                     @override
                                                    Public string to string(){
                                                     +","+feebal+","+timings+","+branch;
//setter and getter
@override
Public string to string(){
sid+","+sname+","+city+","+status=","+totalfee;
}
```

Java Training Center (No 1 in Training & Placement)

WeekendStudent.java	Hibernate.cfg.xml
Package com.jtcindia.hibernate;	<pre><hibernate-configuration></hibernate-configuration></pre>
Import javax.persistence.*;	<session-factory></session-factory>
/*	Session factory,
*@Author:Som Prakash Rai	<mapping< td=""></mapping<>
*@Company:Java Training Center	class="com.jtcindia.hibernate.student"/>
*@See :www.jtcindia.org	<pre><mapping< pre=""></mapping<></pre>
**/	Class="com.jtcindia.hibernate.currentStudent"/>
@Entity	<pre><mapping< pre=""></mapping<></pre>
@Table(name="westudents")	class="com.jtcindia.hibernate.oldStudent"/>
@primary key joinColumn(name="sid")	<mapping< td=""></mapping<>
Public class weekendStudent extends	class="com.jtcindia.hibernate.oldStudent"/>
currentStudent{	
Private string company;	
Private string email;	Vinocinate Configuration
Private string ctc;	
Tilvate sums etc,	*
Public weekendStudent() {}	7
Tublic weekendstudent() {}	
Public weekend Student (string sname, string city,	
string status,	
Double totalfee, double feebal, string timings, strig	Ignite The Search
branch,	
String company, string email, string ctc){	
Super(sname, city, status, totalfee, feebal, timings,	
branch);	
This.company=company; This.email=email;	
This ata-ata	
This.ctc=ctc; GROWT	H UNBOUND
//Setter and getter	2
Whether and getter	2,
@Override	
Public string to string(){	W. W.
Return super.to	170 SNIM
string()+","+company+","+email+","ctc;	ED DIVI
}	
}	
J	6.3
7.0	

(No 1 in Training & Placement)

Table per class mapping

- o In this mapping, you need to take only one table for the entire super and sub classes.
- o 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
	A			ji -					

Osemail	Osctc	Qualification	Percentage	Yop	Wecompany	Weemail	wectc
205						6.3	
		A .			-0-		

Example with Hibernate core (Jtc7) Hibernate persistence classes 1. Public class student { TM 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 (ROWTH UNBOUND 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

30

(No 1 in Training & Placement)

```
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"/>
               city"/>
              cproperty name="city"/>
             property name="status"/>
             property name="totalfee"type=double"/>
             <subclass name="currentStudent"discrimintor-value="CSTU">
       country name=feebal"type="double"/>
           property name="timing"/>
           property name="branch"/>
       <subclass name="currentStudent"discrimintor-value=" RSTU</p>
           property name="qualification"/>
           property name="percentage"/>
           property name="yoe"type="int"/>
       </ioined-subclass>
       <subclass name="currentStudent"discrimintor-value="WSTU">
           property name=" wcompany"/>
           property name=" wcemail"/>
       <property name=" wctc"type="double"/>
           <subclass name="currentStudent"discrimintor-value="OSTS">
           property name=" ostudents"/>
           property name=" ocemail"/>
       cproperty name=" octc"type="double"/>
       </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

(No 1 in Training & Placement)

```
<subclass name="weekendStudent"discriminator</pre>
Student.hbm.xml
<hibernate-mapping ="com.jtcindia.hibernate">
                                                value=" WSTU">
                                                cproperty name="wcompany"/>
<class name="student"table="itcstudents"</pre>
                                                cproperty name="wcemall"/>
Discriminator-value="STU">
                                                cproperty name="wctc"type="double"/>
<id name="sid"column='sid"type="int">
                                                </subclass>
<generator colass="increment"/>
                                                </subclass>
</id>
                                                <subclass name="oldStudent"discriminator</pre>
<discriminator column="stutype"/>
                                                value="OSTU">
property name="sname"/>
                                                property name="ocompany"/>
cproperty name="city"/>
                                                cproperty name="octc"type="double"/>
property name="status"/>
                                                </subclass>
cproperty name="totalfee"type="double"/>
                                                </class>
                                                </hibernate-mapping>
<subclass name="currentStudent"discriminator</pre>
value="CSTU">
cproperty name="feebal"type="double"/>
property name="timings"/>
cproperty name="branch"/>
<subclass name="regularStudent"discriminator</pre>
value="RSTU">
property name="qualification"/>
property name="percentage"/>
property name="yoe"type="int"/>
</subclass>
                                    GROWTH UNBOUND
```

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

32

```
@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
                                                                       TM
       (a) 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")
           O Use the following for super class and all the sub classes
               @Discriminator value (value = "xx")
```

Java Training Center (No 1 in Training & Placement)

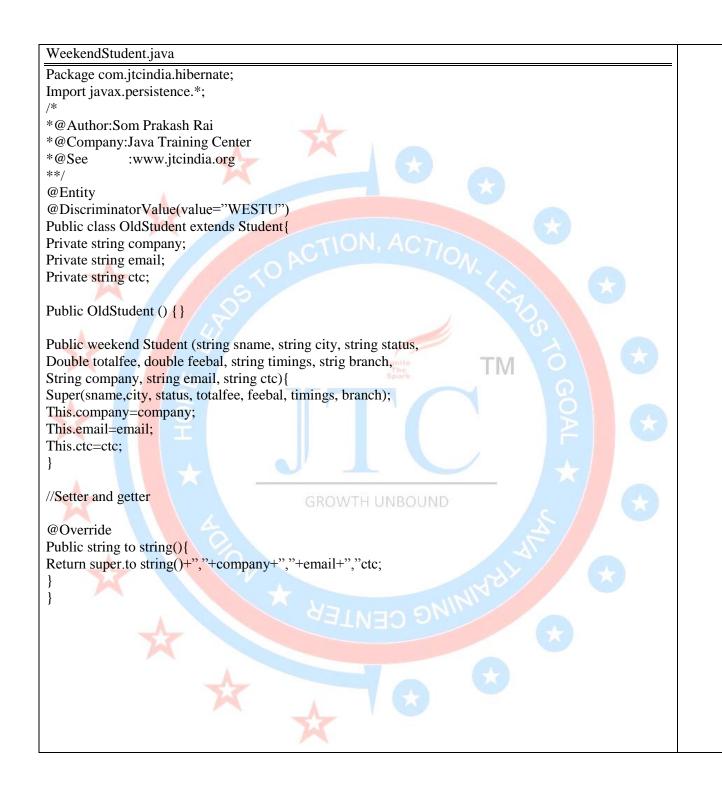
Jtc8: Files required

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	CurrentStudent.java
Package com.jtcindia.hibernate;	Package com.jtcindia.hibernate;
Import javax.persistence.*;	Import javax.persistence.*;
/*	/*
*@Author:Som Prakash Rai	*@Author:Som Prakash Rai
*@Company:Java Training Center	*@Company:Java Training Center
*@See :www.jtcindia.org	*@See :www.jtcindia.org
**/	**/
@Entity	@Entity
@Table(name="jtcstudents")	@DiscriminatorValue(value="CSTU")
@Inheritance(strategy=InheritanceType.SINGLE_TABLE)	Public class currentStudent extends
@DiscriminatorColumn(name="stuType",length=5)	student{
@DiscriminatorValue(value="STU")	@column(name="feebal")
Public class student {	Private double feebal;
@id	@column(name="timings")
@Column(name="sid")	Private string timings;
@GeneratedValue(strategy=generationType.AUTO)	@private string branch")
Private int sid;	Private string branch;
@column(name="sname")	
Private string sname; GROWTH UNB	Public currentStudent(){
@Column(name="city")	}
Private string city	
@Column(name="status")	Public currentStudent (string sname, string
Private string status;	city, string status,
@Column(name="totalfee")	Double totalfee, double feebal, string
Private double totalfee;	timings, string branch){
ASTNEO	Super(sname, city, status, totalfee);
Public student() {}	This. Feebal=feebal;
	This.timings=timings;
Public student (string sname, string city, string status,	This.branch=branch;
double totalfee) {	}
This.sname=sname;	
This.city=city;	//setter and getter
This status = status;	@override
This. Totalfee=totalfee;	Public string to string(){
}	+","+feebal+","+timings+","+branch;
	}
//setter and getter	}
@override	

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

```
OldStudent.java
                                                     WeekdayStudent.java
Package com.jtcindia.hibernate;
                                                     Package com.jtcindia.hibernate;
Import javax.persistence.*;
                                                     Import javax.persistence.*;
*@Author:Som Prakash Rai
                                                     *@Author:Som Prakash Rai
*@Company:Java Training Center
                                                     *@Company:Java Training Center
             :www.jtcindia.org
                                                                   :www.jtcindia.org
*@See
                                                     *@See
**/
                                                     **/
@Entity
                                                     @Entity
@DiscriminatorValue(value="OSTU")
                                                     @DiscriminatorValue(value="WDSTU")
Public class OldStudent extends Student{
                                                     Public class weekdayStudent extends
                                                     CurrentStudent {
Private string company;
Private string email;
                                                     Private string qualificatio;
Private string ctc;
                                                     Private string percentage;
                                                     Private string yop;
Public OldStudent () {}
                                                     Public WeekdayStudent () {}
Public weekend Student (string sname, string city,
string status,
                                                     Public weekend Student (string sname, string city,
Double totalfee, double feebal, string timings, strig
                                                     string status,
branch.
                                                     Double totalfee, double feebal, string timings, strig
String company, string email, string ctc){
                                                     branch.
Super(sname, city, status, totalfee, feebal, timings,
                                                     String company, string email, string ctc){
branch):
                                                     Super(sname, city, status, totalfee, feebal, timings,
This.company=company;
                                                     branch):
                                                     This. qualificatio = qualificatio;
This.email=email;
This.ctc=ctc;
                                                     This. percentage = percentage;
                                                     This. yop = yop;
//Setter and getter
                                                     //Setter and getter
@Override
Public string to string(){
                                                     @Override
Return super.to
                                                     Public string to string(){
                                                     Return super.to string()+","+qualificatio +","+
string()+","+company+","+email+","ctc;
                                                     percentage +"," yop;
}
```



(No 1 in Training & Placement)

3. Table per concrete class mapping

- o 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

				reduce M. Vil	
Sid		Sname	City The Spark	Status	Totalfee
		7			
	7				
	7			28	

2) Cstudents

Sid	Sname	Sity	Status A/T	Totalfee	FEEBAL	TIMINGS	BRANCH
						2	
						Y/ /	

3) Ostudents

Sid	Sname	Sity	Status	Totalfee	Oscompany	Osemail	Osctc
		Z	11-	- 0	111/2		

4) Wdstudents

Sid	Sname	City	Status	Totalfee	Feebal	Timings	Branch	Qualification	Percentage

5) Westudents

Sid	Sname	City	Status	Totalfee	Feebal	Timings	Branch	Wecompany	Weemail
					A				

Java Training Center (No 1 in Training & Placement)

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 jove	
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 voe:	
GROWTH UNBOUND	
Weekend student.java	
Public class Weekday student extends Current student {	
Private string wcompany;	
Private string wcemail; Private double wctc;	
Filvate double wetc,	

38

Java Training Center Mapping Document <hibernate-mapping package="com.jtcindia.hibernate"> (No 1 in Training & Placement) <class name="Student"table="mystudents"> <id name="sid"column="sid"type="int"> <generator coass="increment"/> </id> <discriminator column="stutype"/> cproperty 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>cproperty name="sname"/> city"/> property name="status"/> cproperty 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> coperty name="sname"/> property name="city"/> property name="status"/> property name="totalfee"type=double"/> company"/> cproperty name="ocemail"/> operty name="octc"type="double"/> UNBOUND </class> <class name="regularStudent"table="rstudents1"> <id name="sid"column="sid"type="int"> <generator class="increment"/> </id> property name="sname"/> cproperty name="city"/> property name="status"/> cycle="totalfee"type=double"/> cproperty name="feebal"type="double"/> cproperty name="timings"/> property name=" branch"/> property name="qualification"/> property name="percentage"/> cproperty name="yoe"type="int"/>

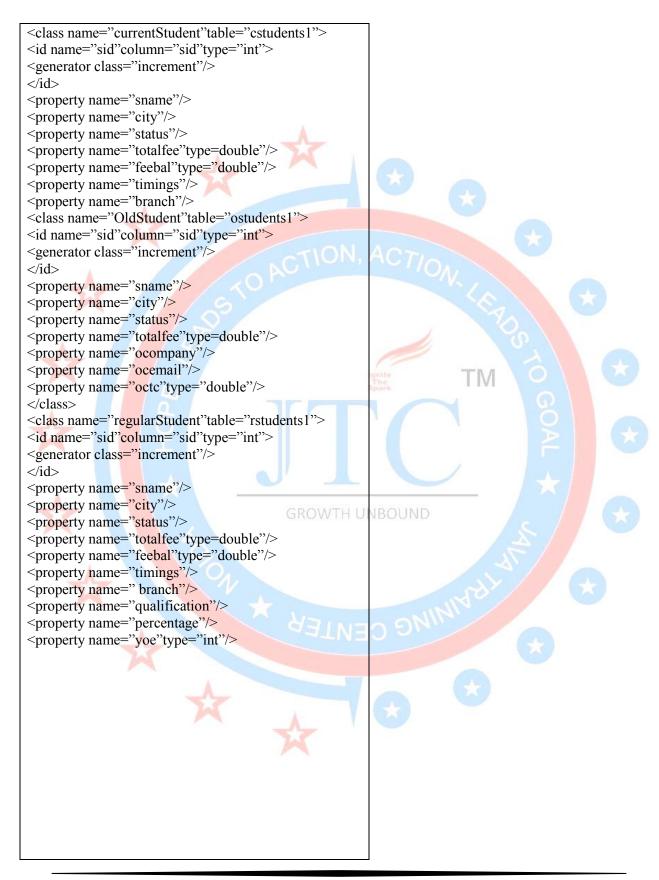
(No 1 in Training & Placement)

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

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 GROWT	Hibernate. Cfg. xml

```
<hibernate-mapping
                                                      </class>
package="com.jtcindia.hibernate">
                                                      <class name="weekedStudent":table="wstudents1">
 <class name="Student"table="mystudents">
                                                      <id name="sid"column="sid"type="int">
 <id name="sid"column="sid"type="int">
                                                      <generator class="increment"/>
 <generator coass="increment"/>
                                                      </id>
 </id>
                                                      cproperty name="sname"/>
 <discriminator column="stutype"/>
                                                      property name="city"/>
 cproperty name="city"/>
                                                      property name="status"/>
                                                      cproperty name="totalfee"type=double"/>
 property name="status"/>
                                                      property name="feebal"type="double"/>
 cproperty name="totalfee"type=double"/>
                                                      cproperty name="timings"/>
 </class>
                                                      property name=" branch"/>
                                                      cproperty name="wcompany"/>
                                                      cproperty name="wcemail"/>
                                                      property name="yoe"type="double"/>
```



(No 1 in Training & Placement)

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 {
```

42

Hibernate Core/Annotation(P1) Author:Som Prakash Rai

www.jtcindia.org

(No 1 in Training & Placement)

}

Note:

- o 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	

GROWTH LINBOLING

Jtc10A.java	Jtc10B.java
Package com.jtcindia.hibernate;	Package com.jtcindia.hibernate;
Import org.hibernate.*;	Import org.hibernate.*;
Public class lab10A{	Public class lab10A{
Public static void main(string[]args){	Public static void main(string[]args){
Transaction tx=null;	Transaction tx=null;
Try{	Try{
SessionFactory	SessionFactory
sf=AhibernateUtil.getSessionFactory();	sf=AhibernateUtil.getSessionFactory();
Session session=sf.openSession();	Session session=sf.openSession();
Tx=session.begintransaction();	Tx=session.begintransaction();
Student st=new	CurrentStudent cs =new
student("som", "noida", "Active", "14000);	CurrentStudent("Rai"."noida","
Session.sava(st);	Active",17000,5000,7:00-9:00","Sec-2");
Tx.commit();	Cs.setSid(1);
Session.close();	Session.close();

```
}catch(exception e) {
}catch(exception e) {
e.printStackTrace();
                                                            e.printStackTrace();
tx.rollback();
                                                            tx.rollback();
Jtc10C.java
                                                            Jtc10D.java
Package com.jtcindia.hibernate;
                                                            Package com.jtcindia.hibernate;
Import java.util.*;
                                                            Import org.hibernate.*;
*@Author:Som Prakash Rai
                                                            *@Author:Som Prakash Rai
*@company:java Training Center
                                                            *@company:java Training Center
*@See
            : www.jtcindia.org
                                                            *@See
                                                                         : www.jtcindia.org
**/
                                                            **/
Public class Jtc10C{
                                                            Public class Jtc10D{
Public static vaid main(string{} args){
                                                            Public static vaid main(string{} args){
Transation tx=null;
                                                            Transation tx=null;
Try{
                                                            Try{
sessionFactory sf=Ahibernate Util.get Session Factory();
                                                            sessionFactory sf=Ahibernate Util.get Session
session session=sf.openSession();
                                                            Factory();
tx=session.beginTransaction();
                                                            session session=sf.openSession();
                                                            tx=session.beginTransaction();
oldStudent os=new
oldStudent("Rai1","noida","Active",15000,"JTC","Prakash
                                                            Weekday student wds=new weekday
@jtc.org","5.6L");
                                                            student('SP","Noida","active",14000,"10:00-
                                                            2:00","NOIDA","Mtech","98%","1999");
Os.setSid(1);
                                                            Os.setSid(1);
Session.sava(os);
Tx.commit();
                                                            Session.sava(os);
}catch(exception e) {
                                                            Tx.commit();
e.prinStackTrace();
                                                            }catch(exception e) {
                                                            e.prinStackTrace();
tx.rollback();
                                                            tx.rollback();
```

Java Training Center (No 1 in Training & Placement)

Jtc10E.java	Jtc10F.java
Package com.jtcindia.hibernate;	Package com.jtcindia.hibernate;
Import org.hibernate.*;	Import org.hibernate.*;
/*	/*
*@Author:Som Prakash Rai	*@Author:Som Prakash Rai
*@company:java Training Center	*@company:java Training Center
*@See : <u>www.jtcindia.org</u>	*@See : <u>www.jtcindia.org</u>
**/	**/
Public class Jtc10D{	Public class Jtc10D{
Public static vaid main(string{} args){	Public static vaid main(string{} args){
Transation tx=null;	Transation tx=null;
Try{ sessionFactory sf=Ahibernate Util.get Session Factory();	Try{ sessionFactory sf=Ahibernate Util.get Session
session session=sf.openSession();	Factory();
tx=session.beginTransaction();	session session=sf.openSession();
	tx=session.beginTransaction();
Weekday student wds=new	
Weekday student(student","Noida"," active", 20000,5000"	Student st=session.load(weekendStudent.class,1);
8:00-2:00","Sec-2" <mark>,"12.</mark> 5L");	System.out.printin(st);
Os.setSid(1);	Session.sava(os);
Sparing constant	Tx.commit();
Session.sava(os); Tx.commit();	}catch(exception e) { e.prinStackTrace();
catch(exception e) {	tx.rollback();
e.prinStackTrace();	tx.ronoack(),
tx.rollback();	
}	
GROWTH UNB	OLIND
} A GROWTH GIVE	OSIND A STATE OF THE STATE OF T
	1140
	ONIN
	311
	**
	★
W	

```
CurrentStudent.java
Student.java
Package com.jtcindia.hibernate;
                                                           Package com.itcindia.hibernate;
Import javax.persistence.*;
                                                           Import javax.persistence.*;
                                                           *@Author:Som Prakash Rai
*@Author:Som Prakash Rai
*@Company:Java Training Center
                                                           *@Company:Java Training Center
*@See
             :www.jtcindia.org
                                                           *@See
                                                                         :www.jtcindia.org
**/
                                                           **/
                                                           @Entity
@Entity
@Table(name="my students1")
                                                           @Table(name="cstudents1")
@Inheritance(strategy=InheritanceType.JOINED)
                                                           @primaryKeyJoinColumn(name="sid")
                                                           Public class currentStudent extends student{
Public class student{
@id
                                                           @column(name="feebal")
@Column(name="sid")
                                                           Private double feebal;
@GeneratedValue(strategy=generationType.AUTO)
                                                           @column(name="timings")
Private int sid:
                                                           Private string timings;
@column(name="sname")
                                                           @private string branch;
Private string sname;
                                                           Private string branch;
@Column(name="city")
Private string city
                                                           Public currentStudent(){
@Column(name="status")
Private string status;
@Column(name="totalfee")
                                                           Public currentStudent (string sname, string city, string
Private double totalfee;
                                                           Double totalfee, double feebal, string timings, string
Public student() {
                                                           branch){
                                                           Super(sname, city, status, totalfee);
                                                           This. Feebal=feebal;
Public student (string sname, string city, string status,
                                                           This.timings=timings;
double totalfee) {
                                                           This.branch=branch;
This.sname=sname;
This.city=city;
This status = status;
                                                           //setter and getter
This. Totalfee=totalfee;
                                                           @override
                                                           Public string to string(){
                                                           +","+feebal+","+timings+","+branch;
//setter and getter
@override
Public string to string(){
Return sid+","+sname+","+city+","+status=","+totalfee;
}
```

Java Training Center (No 1 in Training & Placement)

oldStudent.java	WeekendStudent.java
Package com.jtcindia.hibernate;	Package com.jtcindia.hibernate;
Import javax.persistence.*;	Import javax.persistence.*;
/*	/*
*@Author:Som Prakash Rai	*@Author:Som Prakash Rai
*@Company:Java Training Center	*@Company:Java Training Center
*@See :www.jtcindia.org	*@See :www.jtcindia.org
**/	**/
@Entity	@Entity
@Table(name="ostudents1")	@Table(name=''westudents'')
Public class weekendStudent extends currentStudent{	Public class weekendStudent extends currentStudent{
Private string company;	Private string qualification;
Private string email;	Private string percentage;
Private string ctc;	Private string yop;
AC I Consti	
Public weekendStudent() {}	Public weekendStudent() {}
A / S	
Public weekend Student (string sname, string city, string	Public weekend Student (string sname, string city,
status,	string status,
Double totalfee, double feebal, string timings, strig ctc,	Double totalfee, double feebal, string timings, strig
String company, string email, string ctc){	branch,
Super(sname,city, status, totalfee);	String company, string email, string ctc){
This.company=company;	Super(sname, city, status, total fee, feebal, timings,
This.email=email;	branch);
This.ctc=ctc;	This. qualification = qualification;
}	This. percentage = percentage;
	This. yop = yop;
//Setter and getter	}
GROWTH UNB	OUND
@Override	//Setter and getter
Public string to string(){	
Return super.to	@Override
string()+","+company+","+email+email+","ctc;	Public string to string(){
	Return super.to string()+","+quallfication +","+
	percentage +"," yop;
CENTER	
	*
	*
<u> </u>	

