DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SUBJECT CODE: 19CS2107
ENTERPRISE PROGRAMMING WORKBOOK

HIBERNATE-1#8

Date of the Session:__/__/___ **Time of the Session:** ___to___

Prerequisite:

- Basic idea on Hibernate

Pre-Lab Task:

1. What is ORM tool?

Ans: ORM stands for object-Relational Mapping is a Programming technique for converting data blw relational database & oop languages & java, C# etc.. An ORM tool simplifies data Creation, Manupulation & data access.

2. Disadvantages of JDBC/ Advantages of Hibernate/ Difference between the JDBC and Hibernate

JDBC	Hibernate
It needs to write code to map the object models data to relational and its corresponding schema.	It self maps java classes to database table using xml or by using annotations
able to need to be create manual in database	automatic table creation takes Place.
try-catch blocks are required to handle sql exceptions	need of try-catch blocks, as hibernate handles by converting jdbc to exceptions to unchecked exceptions

3. Name the 4 layers of hibernate architecture

Ans: The four layers of hibernate architecture

(i) java application layer.
(ii) Hibernate framework layer.
(iii) Backend api layer.
(iv) Database layer.

4. Write the syntax of mapping of xml and configuration of xml?

Syntax :

<hibernate-mapping>.
<class name = "POJO class" table = "tablename">.
<id name = "Variable name" column = "column name in database"/>
<Property name = "Variable 1 name" column = "Column name in database"
type = "java/hebernate type"/>.
<Property name = "Variable 2 name" column = "column name in database"
type = java/ hibernate type"/>.
</class> </hibernate-mapping>

Syntax :

<hibernate-configuration>
<session-factory>.
<Property name = "connection.driver class" > Driver class name </Property>.
<Property name = "connection.url" > URL </Property>.
<Property name = "connection.user" > user </Property>.
<Property name = "connection.password > password </Property>.
<Property name = "showsql" > true/ false </Property>.
<Property name = "dealet" > Database dialet class </Property>
<Property name = "hbm2ddl.auto" > create/update </Property>.
<mapping resource = "hbm file 1 name.xml">.
<mapping resource = "hbm file 2 name.xml">.
</session factory>.
</hibernate-Configuration>

In Lab Task:

1. Sreenivas, the proprietor of a rice mill, needs to keep up the information about the rice bags produced at his mill. He wants to save the cost, type of the rice (polished/ non-polished) and amount in kilograms for each bag. Write a hibernate application to insert the details of bags manufactured. The application should ask the user whether he wants to insert a details of a bag each time until he says 'no'. For every 'yes' it should gather the details of the bag i.e, the id number, amount, cost and type.

Writing space for the Problem:(For Student's use only)

Pojo file (Rice bag.java)

```
Package myPackage;
Public class Rice bag{
    Private int id; Private quantity; Private double cost;
Private String type;
  Public int getId(){.
  return id;
  }.
Public void SetId(int id){return quantity;}.
Public int getQuantity(){return quantity;}.
Public void Set quantity{int quantity){this.quantity = quantity}.
Public double get Cost(){return this.cost;}
Public void set Cost(){this.cost = cost;}.
Public String get type(){return type;}
Public void Set Type(){this Type = type}
}
```

Mapping file → Package myPackage;

```
Public class Ricebags{.
<hibernate - mapping>.
<class name="Ricebag.java" table="Ricebag">.
```

<id name="id" column="B-id">

<Property name="quantity" column="quantity" type="String" />

<Property name="Cost" column="Cost type="String" >

</class></hibernate mapping>

Config · file :-

<?xml version="1.0" encoding="UTF-8"?>.

<!DOCTYPE hibernate - Configuration PUBLIC -//Hibernate/ Hibernate Configuration DTD 3.0 //EN" "http://hibernate. Source Forge net/hibernate-Configuration -3.0 dtd">

<Session -factory>.

<Property name="Connection .driver Class" > oracle -jdbc.driver. oracle Driver </Property>.

<Property name="Connection.url"> Connection : dbc : oracle thin: @localhost - : 1552 : xe </Property>.

<Property name="Connection Password" > manager </Property>

<Property name="show-SQL" >true</Property>

<Property name=<hbm2ddl. auto") > update </Property>

<Property name="showSql"> true </Property>

<Property name =dialect "> org. hebernate. dialet. oracle Dialet </Property>

<mapping Resource="riteorg. hbm.xml" />

<session -factory>.

</hibernate Configuration>

Logic file :- Package my Package;

import java.util.*; import java.io.*; import org.hibernate.*;

Writing space for the Problem:(For Student's use only)

```
Public class Details{.
    Public Static void main(String [] args ).
        {
        Scanner SC=new Scanner (System.in);
        Configuration Cfg = new Configuration();
         Cfg.Configure ("hibernate.cfg.xml");
         SessionFactory Sf = Cfg.buildSessionFactory();
        }.
While (Continues)
{ Ricebag r =new Ricebag();
 System.out.Println("enter id of rice bag");
  int amt ,bid =Sc.nextInt();
 System.out.Println("enter quantity of rice bag in kgs);
  int amt = Sc.nextInt();
 System.out.Println("enter cost of ricebag");
 double Cst=Sc.nextDouble();
 System.out.Println("enter type of ricebag");
  String t=Sc.next();
 r.SetId(bid); r.setQuantity(amt); r.SetCost(Cst);
  r.settype(t);
  S.Save(r);
Transaction tx=S.beginTransaction();
 tx.commit();
System.out.Println("Do you want to insert (Yes/No):");
 String Proceed =Sc.next();
if (Proceed.equals("Yes"))
 {
  continues =true;
 }
```

```
else {
    continues = false;
    }
System.out.println("Data has been entered");
  s.close();
}
```

2. Now Sreenivas, wants to sell a bag to a customer. he knows the id number of the bag and needs to retrieve the cost. Sreenivas doesn't encourage bargaining. So if the customer wants to buy the bag at the same value then he sells the bag and erases information about that bag. Write the hibernate application which asks the id number of the bag, displays the price of that bag, and then asks whether the customer wants to purchase the bag or not. If 'yes' then the details of the bag should be erased. (Note: Write only the java file with logic. While executing, implement the POJO, mapping, configuration files of previous question, since we are operating on same data)

Writing space for the Problem:(For Student's use only)

```
import java.util.*; import java.io.*; import org.hibernate.*;
Public class SellBag{
  Public Static void main (String [] args).
  {
    Scanner sc=new Scanner (System.in);
Configuration cfg = new Configuration ();
    Cfg.Configure ("hibernate.Cfg.xml");
    Session factory SF = Cfg.build session factory();
     Session S = SF.open session ();
   System.out.Println("enter id of ricebag");.
  int x = sc.next Int ();
Object o = S.load (Rice bag.class, new Integer(x));
Ricebag r = (Ricebag)o;
System.out.Println (" the cost of ricebag is "+ r.getCost ());
System out.Print ln("want to Purchase bag (yes/no)");
String y = sc.next ();
  if (y.equals("yes")){
        Transaction tx = S.begin Transaction();
         S.delete (r);
         tx.Commit();
   System.out.Print ln ("Bag is Sold");.
  }
else {.
```

Writing space for the Problem:(For Student's use only)

```
System.out.Println("a lot Sold");
}
s.close();
}
}
```

Post Lab Task:

1. Gokul Fabrics has wide range of various fabrics in different colours. The owner of the shop needs to keep up the information of the fabrics he have. He want to save the type, colour, available length and cost per metre for each fabric. He also wants to update the length of the fabric when he sells it to the customer.
   a) Write a Hibernate application to insert the details of the fabric.
   b) Write a Hibernate application which requests the id number of the fabric and its length sold, displays the total selling cost and then updates the data with the remaining length.

Writing space for the Problem:(For Student's use only)

```
Package myPackage;
Public class Gokul Fabrics {.
    Private int bid;
    Private String type;
    Private double cost Permeter;.
    Private double availability;.
    Private double cost Per meter;
    Public int getFid() { return fid; }.
    Public String get type { return type; }.
    Public void Set Fid (int Fid) { this.Fid = fid; }.
    Public void set type (String type) { this.type = type }.
    Public void Set AVailability (double availability) {.
        this.availability = availability;
    }
    Public void set Cost Per meter (double cost Permeter) {.
        this Cost Per meter = Cost Per meter;
    }
}.
```

Config mapp file:-

```
<?xml version="1.0" encoding "UTF-8"?>
<!DOC type hibernate-mapping Public "//Hibernate //Hibernate mapping
DTD 3.0//EN" " http://hibernate Source forge.net/ hibernate-mapping-3.0
```

<Hibernate - Configuration>.

<Session - Factory>.

<Property name = "Connection.driver - class"> oracle.jdbc.driver.

Oracle Driver </Property>

<Property name = "Connection url"> jdbc:oracle:thin:localhost:

1552:xe </Property>.

<Property name = "Connection.user"> System </Property>.

<Property name = "Connection.password"> manager </Property>.

<Property name = "Show-SQL"> true </Property>

<Property name = "hbm2ddl.auto"> update </Property>.

<Property name = "dialect"> org.hibernate.dialect.Oracle9Dialect

</Property>.

<mapping resource = "gokulfabrics.hbm.xml" />

</Session-Factory

</hibernate - Configuration>

Mapping file :-

<?xml version = "1.0" encoding = "UTF-8"?>

<hibernate-mapping>

<class name = "myPackage.GokulFabrics" table = "Fabrics">.

<id name = "fid" column = "id no"> <generator class = "assigned">

</id>.

<Property name = "type" Column type" />.

<Property name = "availability" Column = "length" />

<Property name = "Cost Parameter" column = "cost" />.

</class>

</hibernate-mapping>.

Writing space for the Problem:(For Student's use only)

Logic File:-

```
Package myPackage;
import java.util.*; import org.hibernate.*;
import myPackage:Rice bag;
Public class Insert Fabric {
    Public Static void main (String [] args).
    {
Scanner Sc = new Scanner (System.in);
 Bodean Continues = true;.
Configuration Cfg = new Configuration ();
Cfg. Configure ("hibernate.hbm.xml");
Session Factory SF = Cfg.build session Factory ();
   Session S = SF.open Session ();.
While (Continues)
{
Gokul Fabrics g = new Gokul Fabrics ();
System.out.Print ln ("Enter id of Fabric");
int id = SC.next Int ();
System.out.Print ln ("enter type of Fabric");
   String type = SC.next();
System.out.Print ln ("enter availability of Fabric in meters:");
double avail = SC.next Double();
System.out.Print ln ("enter cost Per meter");
Double C = SC.next Double ();
   g.Self Id x;
   g.set Type (type);
   g.Set Availability(avail)
   S.Save (g);
Transaction tx = S.begin Transactions
```

```
tx.Commit();
System.out.Println("Do you want to insert (Yes/no):");
String Proceed = Sc.next();
if (Proceed.equals("yes")).
    {
    Continues = true;
    }
else
    Continues = false;
System.out.Println("Data has been entered");
S.close(); } }.
```

```
Package myPackage;
import java.util.*; import java.io.*, import org.hibernate.*;
Public class update Fabric {.
    Public static void main (String[] args).
{
    Scanner sc = new Scanner (System.in);
    System.out.Println("enter id of fabric");
    int x = SC.nextInt();
    System.out.Println("enter length of fabric");
    double y = SC.next.Double();
    Configuration cfg = new Configuration
      Cfg.Configure("hibernate Cfg.xml");
    Session Factory Sf = Cfg.build Session Factory();
      Session S = Sf.open Session();
      object o = S.load (Gokul Fabrics.class, new Integers(x));
    Gokul Fabrics g = (Gokul Fabrics) o;
    System.out.Println("The Selling Cost of fabric is :" + Selling Cost);
      g.Set Availability (rem);
      tn.Commit();
```

Writing space for the Problem:(For Student's use only)

```
System.out.Println("Data is Uploaded");
S.close();
}
}
```