**------------------------------------1------------------------------------**

**Prime number program**

package collection;

//. Create an ArrayList al and add 25 random numbers. Write a code to print all the prime numbers that are present in it, using lambda expression.

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.List;

interface Prime{

void checkPrime(List<Integer> number);

}

public class PrimeNumber {

public static void main(String args[])throws IOException {

List<Integer> listOfNumber=new ArrayList<Integer>();

//adding ranraom values to list

BufferedReader bufferedReader=new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter 25 random numbers : ");

for(int i=0;i<5;i++) {

listOfNumber.add(Integer.parseInt(bufferedReader.readLine()));

}

System.out.println("All the prime numbers are");

Prime prime=(numbers)->{

for(int j=0;j<5;j++) {

int flag=0;

for(int i=2;i<=listOfNumber.get(j)/2;i++) {

if((listOfNumber.get(j)%i)==0) {

flag=1;

break;

}

}

if(flag==0) {

System.out.println(listOfNumber.get(j)+" is prime number");

}

}

};

prime.checkPrime(listOfNumber);

}

}

**----------------------------------End 1-------------------------------**

**-----------------------------2------------------------------------------**

**Employee.java**

**package** com.employeee;

**public** **class** Employee {

**private** Integer empNo;

**private** String name;

**private** Integer age;

**private** String location;

**public** Employee(Integer empNo, String name, Integer age, String location) {

**super**();

**this**.empNo = empNo;

**this**.name = name;

**this**.age = age;

**this**.location = location;

}

**public** Integer getEmpNo() {

**return** empNo;

}

**public** **void** setEmpNo(Integer empNo) {

**this**.empNo = empNo;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** Integer getAge() {

**return** age;

}

**public** **void** setAge(Integer age) {

**this**.age = age;

}

**public** String getLocation() {

**return** location;

}

**public** **void** setLocation(String location) {

**this**.location = location;

}

}

**ManageEmployee.java**

package com.employeee;

//.Create an Employee class with

//1. Instance variables: empNo, name, age, location.

//2. A parameterized constructor to initialize them.

//Write a program using streams

//1. To add five Employee objects into an ArrayList,

//2. Filter the Employee objects whose location is Pune,

//3. Store them in a separate ArrayList and print their details

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.ArrayList;

import java.util.Iterator;

import java.util.List;

public class ManageEmployee {

public static void main(String args[])throws IOException {

//Variables

Integer empNo;

String name;

Integer age;

String location;

//List of employee

List<Employee> listOfEmployee=new ArrayList<Employee>();

//Taking input from user

BufferedReader bufferedReader=new BufferedReader(new InputStreamReader(System.in));

System.out.println("\nEnter employee 5 records:");

for(int i=1;i<=3;i++) {

System.out.println("Enter "+i+" employee detils :");

System.out.print("Enter employee no. : ");

empNo=Integer.parseInt(bufferedReader.readLine());

System.out.print("Enter employee name : ");

name=bufferedReader.readLine();

System.out.print("Enter employee age : ");

age=Integer.parseInt(bufferedReader.readLine());

System.out.print("Enter employee location : ");

location=bufferedReader.readLine();

listOfEmployee.add(new Employee(empNo, name, age, location));

}

//filttering record using condition from list

System.out.println("The details of employee whose location is pune ");

List<Employee> puneEmployee=

listOfEmployee.stream()

.filter(EmployeeLocation-> EmployeeLocation.getLocation().equalsIgnoreCase("Pune"))

.toList();

//Traversing listOfEmployee

Iterator<Employee> iterator=puneEmployee.iterator();

while(iterator.hasNext()) {

Employee employee=iterator.next();

System.out.println(employee.getEmpNo()+" "+employee.getName()+" "+employee.getAge()+" "+employee.getLocation());

}

}

}

**------------------------------End 2------------------------------**

**------------------------------3------------------------------------**

**Optional example**

RemoveNullPointerException.java

package OptionalExample;

//2.Use Optional class and avoid NullPointerException from the below code:

**//String names[] = new String[5];**

//System.out.print(names[0].length( ));

import java.util.Optional;

public class RemoveNullPointerException {

public static void main(String args[]) {

String names[] = new String[5];

Optional<String> checkNull=Optional.ofNullable(names[4]);

if(checkNull.isPresent()) {

System.out.print(names[0].length());

}

else {

System.out.println("String array is empty");

}

}

}

**-------------------------------End 3-------------------------------**

**-------------------------Databse Mapping--------------------**

**BillingAddress.java**

package com.entity;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OrderColumn;

import javax.persistence.Table;

@Entity

@Table(name="BillingAddress")

public class BillingAddress {

@Id

@GeneratedValue(strategy =GenerationType.AUTO)

private Integer billingAddress\_id;

private String address;

private String city;

private String state;

private Integer zipcode;

private String country;

@OneToMany(cascade=CascadeType.ALL)

@JoinColumn(name="billingAddress\_id")

@OrderColumn(name="type")

private List<Customer> customer;

@OneToMany(cascade=CascadeType.ALL)

@JoinColumn(name="billingAddress\_id")

@OrderColumn(name="type")

private List<SalesOrder> salesOrders;

//Getters and setters method

public Integer getBillingAddress\_id() {

return billingAddress\_id;

}

public void setBillingAddress\_id(Integer billingAddress\_id) {

this.billingAddress\_id = billingAddress\_id;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getCity() {

return city;

}

public void setCity(String city) {

this.city = city;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public Integer getZipcode() {

return zipcode;

}

public void setZipcode(Integer zipcode) {

this.zipcode = zipcode;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public List<Customer> getCustomer() {

return customer;

**}**

public void setCustomer(List<Customer> customer) {

this.customer = customer;

}

public List<SalesOrder> getSalesOrders() {

return salesOrders;

}

public void setSalesOrders(List<SalesOrder> salesOrders) {

this.salesOrders = salesOrders;

}

}

**Cart.java**

package com.entity;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OrderColumn;

import javax.persistence.Table;

@Entity

@Table(name="Cart")

public class Cart {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer cart\_Id;

private Double totalPrice;

@OneToMany(cascade =CascadeType.ALL)

@JoinColumn(name="cart\_id")

@OrderColumn(name="type")

private List<Customer> customers;

@OneToMany(cascade =CascadeType.ALL)

@JoinColumn(name="cart\_id")

@OrderColumn(name="type")

private List<SalesOrder> salesOrders;

@OneToMany(cascade=CascadeType.ALL)

@JoinColumn(name="cart\_id")

@OrderColumn(name="type")

private List<CartItem> cartItems;

//Getters and setters

public Integer getCart\_Id() {

return cart\_Id;

}

public void setCart\_Id(Integer cart\_Id) {

this.cart\_Id = cart\_Id;

}

public Double getTotalPrice() {

return totalPrice;

}

public void setTotalPrice(Double totalPrice) {

this.totalPrice = totalPrice;

}

public List<Customer> getCustomers() {

return customers;

}

public void setCustomers(List<Customer> customers) {

this.customers = customers;

}

public List<SalesOrder> getSalesOrders() {

return salesOrders;

}

public void setSalesOrders(List<SalesOrder> salesOrders) {

this.salesOrders = salesOrders;

}

}

**CartItem.java**

package com.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="CartItem")

public class CartItem {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer cartItem\_id;

private Integer quantity;

private Double price;

public Integer getCartItem\_id() {

return cartItem\_id;

}

public void setCartItem\_id(Integer cartItem\_id) {

this.cartItem\_id = cartItem\_id;

}

public Integer getQuantity() {

return quantity;

}

public void setQuantity(Integer quantity) {

this.quantity = quantity;

}

public Double getPrice() {

return price;

}

public void setPrice(Double price) {

this.price = price;

}

}

**Customer.java**

package com.entity;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OrderColumn;

import javax.persistence.Table;

@Entity

@Table(name="Customer")

public class Customer {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer Customer\_id;

private String firstName;

private String lastName;

private String customerPnone;

public Customer() {

// TODO Auto-generated constructor stub

}

//Parameterized constructor

public Customer(String firstName, String lastName, String customerPnone) {

super();

this.firstName = firstName;

this.lastName = lastName;

this.customerPnone = customerPnone;

}

@OneToMany(cascade =CascadeType.ALL)

@JoinColumn(name="Customer\_id")

@OrderColumn(name="type")

private List<SalesOrder> salesOrders;

//Getters and setter method

public Integer getCustomer\_id() {

return Customer\_id;

}

public void setCustomer\_id(Integer customer\_id) {

Customer\_id = customer\_id;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public String getCustomerPnone() {

return customerPnone;

}

public void setCustomerPnone(String customerPnone) {

this.customerPnone = customerPnone;

}

public List<SalesOrder> getSalesOrders() {

return salesOrders;

}

public void setSalesOrders(List<SalesOrder> salesOrders) {

this.salesOrders = salesOrders;

}

}

**Product.java**

package com.entity;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OrderColumn;

import javax.persistence.Table;

@Entity

@Table(name="Product")

public class Product {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer product\_id;

private String category;

private String description;

private String name;

private double price;

private Integer unit;

public Product() {

}

public Product(String category, String description, String name, double price, Integer unit) {

super();

this.category = category;

this.description = description;

this.name = name;

this.price = price;

this.unit = unit;

}

@OneToMany(cascade=CascadeType.ALL)

@JoinColumn(name="product\_id")

@OrderColumn(name="type")

private List<CartItem> cartItems;

//Getters and setters method

public Integer getProduct\_id() {

return product\_id;

}

public void setProduct\_id(Integer product\_id) {

this.product\_id = product\_id;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

public Integer getUnit() {

return unit;

}

public void setUnit(Integer unit) {

this.unit = unit;

}

public List<CartItem> getCartItems() {

return cartItems;

}

public void setCartItems(List<CartItem> cartItems) {

this.cartItems = cartItems;

}

}

**SalesOrder.java**

package com.entity;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.Table;

@Entity

@Table(name="SalesOrder")

public class SalesOrder {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer salesOrder\_id;

public Integer getSalesOrder\_id() {

return salesOrder\_id;

}

public void setSalesOrder\_id(Integer salesOrder\_id) {

this.salesOrder\_id = salesOrder\_id;

}

}

**ShippingAddress.java**

package com.entity;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OrderColumn;

import javax.persistence.Table;

import org.hibernate.annotations.Cascade;

@Entity

@Table(name="ShippingAddress")

public class ShippingAddress {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer shippingAddress\_id;

private String address;

private String state;

private Integer zipcode;

private String country;

@OneToMany(cascade=CascadeType.ALL)

@JoinColumn(name="shippingAddress\_id")

@OrderColumn(name="type")

private List<Customer> customer1;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name="shippingAddress\_id")

@OrderColumn(name="type")

private List<SalesOrder> salesOrders;

//Getters and setters

public Integer getShippingAddress\_id() {

return shippingAddress\_id;

}

public void setShippingAddress\_id(Integer shippingAddress\_id) {

this.shippingAddress\_id = shippingAddress\_id;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public String getState() {

return state;

}

public void setState(String state) {

this.state = state;

}

public Integer getZipcode() {

return zipcode;

}

public void setZipcode(Integer zipcode) {

this.zipcode = zipcode;

}

public String getCountry() {

return country;

}

public void setCountry(String country) {

this.country = country;

}

public List<Customer> getCustomer1() {

return customer1;

}

public void setCustomer1(List<Customer> customer1) {

this.customer1 = customer1;

}

public List<SalesOrder> getSalesOrders() {

return salesOrders;

}

public void setSalesOrders(List<SalesOrder> salesOrders) {

this.salesOrders = salesOrders;

}

}

**User.java**

package com.entity;

import java.util.List;

import javax.persistence.CascadeType;

import javax.persistence.Entity;

import javax.persistence.GeneratedValue;

import javax.persistence.GenerationType;

import javax.persistence.Id;

import javax.persistence.JoinColumn;

import javax.persistence.OneToMany;

import javax.persistence.OrderColumn;

import javax.persistence.Table;

import javax.persistence.criteria.CriteriaBuilder.In;

@Entity

@Table(name="User")

public class User {

@Id

@GeneratedValue(strategy = GenerationType.AUTO)

private Integer user\_id;

private String emailid;

private String password;

private String enabled;

@OneToMany(cascade = CascadeType.ALL)

@JoinColumn(name="user\_id")

@OrderColumn(name="type")

private List<Customer> customers;

public Integer getUser\_id() {

return user\_id;

}

public void setUser\_id(Integer user\_id) {

this.user\_id = user\_id;

}

public String getEmailid() {

return emailid;

}

public void setEmailid(String emailid) {

this.emailid = emailid;

}

public String getPassword() {

return password;

}

public void setPassword(String password) {

this.password = password;

}

public String getEnabled() {

return enabled;

}

public void setEnabled(String enabled) {

this.enabled = enabled;

}

public List<Customer> getCustomers() {

return customers;

}

public void setCustomers(List<Customer> customers) {

this.customers = customers;

}

}

**CustomerServices.java**

package com.hibernate.services;

import java.util.Iterator;

import java.util.List;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.Transaction;

import com.entity.Customer;

import com.entity.Customer;

import com.hibernate.util.HibernateUtil;

public class CustomerServices {

static Session sessionObj;

/\*------------------------Post request--------------------------------------\*/

/\* Method to store data in database business logic \*/

public Integer addCustomer(String firstName, String lastName, String customerPnone) {

Transaction transaction=null;

Integer CustomerDetails=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

Customer customer=new Customer(firstName, lastName, customerPnone);// created the object of Customer class

CustomerDetails=(Integer)sessionObj.save(customer);// save the object or insert the recording

transaction.commit();// explictiy call the commit esure that auto commite should be false

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

return CustomerDetails;

}

/\*-----------------------------------Get Request-------------------------------\*/

public void listOfAllCustomers() {

System.out.println("All the available Customers");

Transaction transaction=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

//Retrieve logic

List customers=sessionObj.createQuery("From Customer").list();

Iterator<Customer> iterator=customers.iterator();

while(iterator.hasNext()) {

Customer customer1=iterator.next();

System.out.println("Customer id : "+customer1.getCustomer\_id());

System.out.println("Customer Name : "+customer1.getFirstName()+" "+customer1.getLastName());

System.out.println("Phone number : "+customer1.getCustomerPnone());

}

transaction.commit();

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

}

/\*----------------------------------------PUT request---------------------\*/

/\*Method for updating the records in the database\*/

public void updateCustomerDetails(int Customer\_id,String mobilenumber) {

Transaction transaction=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

//logic

Customer customer=(Customer) sessionObj.get(Customer.class, Customer\_id);

customer.setCustomerPnone(mobilenumber);

sessionObj.update(customer);;

System.out.println("Updated sucessfully ");

transaction.commit();

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

}

/\*-------------------------------------Delete Request----------------------\*/

/\*Method for deleteding the Customer \*/

public void deleteCustomerById(Integer CustomerId) {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*deleting record from the database\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Transaction transaction=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

Customer customer=(Customer)sessionObj.get(Customer.class, CustomerId);

sessionObj.delete(customer);// hibernate will form delete query automatically

System.out.println("deleted sucessfully..."+customer.getCustomer\_id());;

transaction.commit();// explictiy call the commit esure that auto commite should be false

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

}

}

**ProductServices.java**

package com.hibernate.services;

import com.entity.Product;

import com.hibernate.util.\*;

import java.util.Iterator;

import java.util.List;

import org.hibernate.HibernateException;

import org.hibernate.Session;

import org.hibernate.SessionFactory;

import org.hibernate.Transaction;

public class ProductServices {

static Session sessionObj;

/\*------------------------Post request--------------------------------------\*/

/\* Method to store data in database business logic \*/

public Integer addProduct(String category, String description, String name, double price, Integer unit) {

Transaction transaction=null;

Integer productDetails=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

Product product=new Product(category, description, name, price, unit);// created the object of product class

productDetails=(Integer)sessionObj.save(product);// save the object or insert the recording

transaction.commit();// explictiy call the commit esure that auto commite should be false

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

return productDetails;

}

/\*-----------------------------------Get Request-------------------------------\*/

public void listOfAllProducts() {

System.out.println("All the available products");

Transaction transaction=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

//Retrieve logic

List products=sessionObj.createQuery("From Product").list();

Iterator<Product> iterator=products.iterator();

while(iterator.hasNext()) {

Product product1=iterator.next();

System.out.println("Product id : "+product1.getProduct\_id());

System.out.println("Product Category : "+product1.getCategory());

System.out.println("Description : "+product1.getDescription());

System.out.println("Product name : "+product1.getName());

System.out.println("Product price : "+product1.getPrice());

System.out.println("Product unit : "+product1.getUnit());

}

transaction.commit();

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

}

/\*----------------------------------------PUT request---------------------\*/

/\*Method for updating the records in the database\*/

public void updateProductDetails(int product\_id,double price) {

Transaction transaction=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

//logic

Product product=(Product) sessionObj.get(Product.class, product\_id);

product.setPrice(price);

sessionObj.update(product);;

System.out.println("Updated sucessfully ");

transaction.commit();

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

}

/\*-------------------------------------Delete Request----------------------\*/

/\*Method for deleteding the product \*/

public void deleteProductById(Integer productId) {

System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*deleting record from the database\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

Transaction transaction=null;

try {

sessionObj=HibernateUtil.buildSessionFactory().openSession();

transaction=sessionObj.beginTransaction();

Product product=(Product)sessionObj.get(Product.class, productId);

sessionObj.delete(product);// hibernate will form delete query automatically

System.out.println("deleted sucessfully..."+product.getProduct\_id());;

transaction.commit();// explictiy call the commit esure that auto commite should be false

}

catch(HibernateException e) {

if(transaction!=null) {

transaction.rollback();

}

e.printStackTrace();

}

finally {

sessionObj.close();

}

}

}

**HibernateUtil.java**

package com.hibernate.util;

import org.hibernate.SessionFactory;

import org.hibernate.boot.Metadata;

import org.hibernate.boot.MetadataSources;

import org.hibernate.boot.registry.StandardServiceRegistry;

import org.hibernate.boot.registry.StandardServiceRegistryBuilder;

import com.mysql.cj.Session;

public class HibernateUtil {

static SessionFactory sessionFactory;

static Session sessionObj;

public static SessionFactory buildSessionFactory() {

StandardServiceRegistry standardServiceRegistryObj=new StandardServiceRegistryBuilder().configure("hibernate.cfg.xml").build();

Metadata metadata=new MetadataSources(standardServiceRegistryObj).getMetadataBuilder().build();

sessionFactory=metadata.getSessionFactoryBuilder().build();

return sessionFactory;

}

}

**ManageRecords.java**

package com.UserInterface;

import java.util.ArrayList;

import java.util.List;

import org.hibernate.id.IntegralDataTypeHolder;

import com.hibernate.services.CustomerServices;

import com.hibernate.services.ProductServices;

public class ManageRecords {

public static void main(String args[]) {

/\*-----------------------------------------Product Crud Operation-----------------------------------------------\*/

ProductServices productServices=new ProductServices();

Integer product1=productServices.addProduct("Clothes", "L size", "Shirt", 320, 5);

List<Integer> listProduct=new ArrayList<>();

listProduct.add(product1);

if(listProduct.size()!=0) {

System.out.println("Record is inserted");

}

else

{

System.out.println("Record is not inserted");

}

/\*List of product records in databse\*/

try {

productServices.listOfAllProducts();

}

catch(Exception e) {

e.printStackTrace();

}

/\*Update product records in database\*/

try {

productServices.updateProductDetails(3, 600);

}

catch(Exception e) {

e.printStackTrace();

}

/\* delete product records in database by id \*/

try {

productServices.deleteProductById(2);

}

catch(Exception e) {

e.printStackTrace();

}

/\*------------------------------------------End of product crud operation-------------------------------------\*/

/\*------------------------------------------Customer crud operation-------------------------------------------\*/

CustomerServices customerServices=new CustomerServices();

Integer customer1=customerServices.addCustomer("Ankit", "Jaisawal", "8299883190");

Integer customer2=customerServices.addCustomer("Amit", "Jaisawal", "8293673190");

List<Integer> listOfCustomer=new ArrayList<>();

listOfCustomer.add(customer1);

listOfCustomer.add(customer2);

if(listOfCustomer.size()!=0) {

System.out.println("Record is inserted ");

}

else {

System.out.println("Record is not inserted ");

}

/\*List of all customer\*/

try {

customerServices.listOfAllCustomers();

}

catch(Exception e) {

e.printStackTrace();

}

/\*Update customer record in database\*/

try {

customerServices.updateCustomerDetails(8, "8290063646");

}

catch(Exception e) {

e.printStackTrace();

}

/\*Delete customer record in database by id\*/

try {

customerServices.deleteCustomerById(6);

}

catch(Exception e) {

e.printStackTrace();

}

/\*------------------------------------------End of crud operation---------------------------------------------\*/

}

}

**Hibernate.cfg.xml**

<?xml version=*'1.0'* encoding=*'UTF-8'*?>

<!DOCTYPE hibernate-configuration PUBLIC

"-//Hibernate/Hibernate Configuration DTD 5.3//EN"

"http://hibernate.sourceforge.net/hibernate-configuration-5.3.dtd">

<hibernate-configuration>

<session-factory>

<property name=*"hbm2ddl.auto"*>create</property>

<property name=*"dialect"*>org.hibernate.dialect.MySQL5Dialect</property>

<property name=*"connection.url"*>jdbc:mysql://localhost/ECommerce</property>

<property name=*"connection.username"*>root</property>

<property name=*"connection.password"*>root</property>

<property name=*"connection.driver\_class"*>com.mysql.jdbc.Driver</property>

<property name=*"show\_sql"*>true</property>

<property name=*"format\_sql"*>true</property>

<property name=*"hbm2ddl.auto"*>update</property>

<mapping class=*"com.entity.Customer"* />

<mapping class=*"com.entity.ShippingAddress"* />

<mapping class=*"com.entity.BillingAddress"* />

<mapping class=*"com.entity.User"* />

<mapping class=*"com.entity.Cart"* />

<mapping class=*"com.entity.SalesOrder"* />

<mapping class=*"com.entity.Product"* />

<mapping class=*"com.entity.CartItem"* />

</session-factory>

</hibernate-configuration>

**Output of database mapping:**















