**Name – Tushar Verma**

**Batch - ERDJA - 1 - B13 – Capgemini**

**Superset id – 1097003**

**Module 1 - Coding Assignment (Core Java 8\_Database & PostGreSQL with DevOps)**

Question 1 solution

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.Set;

import java.util.stream.Collectors;

class Employee {

private int id;

private String name;

private String address;

private double salary;

public Employee() { }

public Employee(int id, String name, String address, double salary) {

this.id = id;

this.name = name;

this.address = address;

this.salary = salary;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

@Override

public String toString() {

return "Employee{" +

"id=" + id +

", name='" + name + '\'' +

", address='" + address + '\'' +

", salary=" + salary +'\''+

'}' +'\'';

}

}

public class Demo {

public static void main(String[] args) {

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

empList.add(new Employee(101,"Raj","Vijayawada",50000));

empList.add(new Employee(102,"Rahul","Hyderabad",100000));

empList.add(new Employee(103,"Manoj","Pune",150000));

empList.add(new Employee(104,"Kajal","Pune",48000));

empList.add(new Employee(105,"Harry","Hyderabad",100000));

System.out.println("\nSorted by salary and name :");

Collections.sort(empList, (e1,e2) ->

(e1.getSalary()!=e2.getSalary()) ? (int) (e1.getSalary() - e2.getSalary()) : e1.getName().compareTo(e2.getName()));

empList.forEach(emp -> System.out.println(emp));

}

}

Question 2 solution

class Student{

private int roll;

private String name;

private int marks;

private String result;

public Student(){ }

public Student(int roll, String name, int marks) {

this.roll = roll;

this.name = name;

this.marks = marks;

this.result = result;

}

public int getRoll() {

return roll;

}

public void setRoll(int roll) {

this.roll = roll;

}

public String getName() {

return this.name;

}

public void setName(String name) {

this.name = name;

}

public int getMarks() {

return this.marks;

}

public void setMarks(int marks) {

this.marks = marks;

}

public String getResult() {

return this.result;

}

public String studentResult(){

if(this.getMarks()>500)

return "Student Result is : Pass";

else

return "Student Result is : Fail";

}

@Override

public String toString() {

return "Student Roll is : " + this.getRoll() + "\nStudent Name is : " + this.getName()

+ "\nStudent Marks is : " + this.getMarks() + "\n" + this.studentResult();

}

}

public class Demo {

public static void main(String[] args){

Scanner sc=new Scanner(System.in);

System.out.println("Enter number of students");

String str=sc.next();

String []arr = str.split(":");

Student std=new Student(Integer.parseInt(arr[0]),arr[1],Integer.parseInt(arr[2]));

System.out.println(std);

}

}

Question 3 solution

import java.util.ArrayList;

import java.util.List;

import java.util.Set;

import java.util.stream.Collectors;

class Employee {

private int id;

private String name;

private String address;

private double salary;

public Employee() { }

public Employee(int id, String name, String address, double salary) {

this.id = id;

this.name = name;

this.address = address;

this.salary = salary;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

public double getSalary() {

return salary;

}

public void setSalary(double salary) {

this.salary = salary;

}

public int hashCode() {

int result = 7 \* 3 + this.getAddress().hashCode();

return result;

}

// equals() method

@Override

public boolean equals(Object obj) {

if (this == obj)

return true;

if (obj == null)

return false;

if (this.getClass() != obj.getClass())

return false;

Employee emp = (Employee) obj;

return this.address.equalsIgnoreCase(emp.getAddress());

}

@Override

public String toString() {

return "Employee{" +

"id=" + id +

", name='" + name + '\'' +

", address='" + address + '\'' +

", salary=" + salary +'\''+

'}' +'\'';

}

}

public class RemovingDuplicates {

public static void main(String[] args) {

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

empList.add(new Employee(101,"Raj","Vijayawada",50000));

empList.add(new Employee(102,"Rahul","Hyderabad",100000));

empList.add(new Employee(103,"Manoj","Pune",150000));

empList.add(new Employee(104,"Kajal","Pune",48000));

empList.add(new Employee(105,"Sam","Hyderabad",180000));

System.out.println("Original Employee list with duplicate addresses :");

empList.forEach(emp -> System.out.println(emp));

Set<Employee> uniqueEmployeeSet = empList.stream().distinct().collect(Collectors.toSet());

System.out.println("\nEmployee list without duplicate addresses :");

uniqueEmployeeSet.forEach(emp -> System.out.println(emp));

}

}

Question 4 solution

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.Statement;

public class Demo {

public static boolean saveEmployeeDetails(int id, String name, String address, double price) {

try{

Class.forName("org.postgres.Driver");

Connection con= DriverManager.getConnection("jdbc:postgres://localhost:5432/demodb","postgres","root");

Statement stmt=con.createStatement();

stmt.executeUpdate("insert into employee VALUES ("+id+", '"+name+"', '"+address+"', "+price+");");

con.close();

return true;

}catch(Exception e){

System.out.println(e);

}

return false;

}

public static void main(String[] args) {

if(saveEmployeeDetails(101,”Kamal","Lucknow",50000))

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

}

}

--------------------------------------------------------------------------------------------------------------------------------------