5. Write a JDBC program to read Employee(name,id,dept,desg,company,emailed) from the user and insert into mysql table called employee.(25 Marks)

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.PreparedStatement;

**import** java.sql.SQLException;

**import** java.util.Scanner;

**public** **class** UserMain5 {

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

**try** {

DriverManager.*registerDriver*(**new** com.mysql.cj.jdbc.Driver());

Connection connection=DriverManager.

*getConnection*("jdbc:mysql://localhost:3306/Employee","root","root");

String querry="insert into employee values(?,?,?,?,?,?)";

PreparedStatement pst=connection.prepareStatement(querry);

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter the records.......");

System.***out***.println("enter name");

String name=sc.next();

System.***out***.println("enter id");

**int** id=sc.nextInt();

System.***out***.println("enter department");

String dept=sc.next();

System.***out***.println("enter designation");

String desg=sc.next();

System.***out***.println("enter company");

String company=sc.next();

System.***out***.println("enter emailed");

String emailed=sc.next();

pst.setString(1,name );

pst.setInt(2 ,id);

pst.setString(3,dept );

pst.setString(4,desg );

pst.setString(5,company );

pst.setString(6,emailed );

**int** x=pst.executeUpdate();

**if**(x!=0)

System.***out***.println("Records Succesfully inserted");

**else**

System.***out***.println("Not inserted");

pst.close();

connection.close();

}

**catch**(Exception e)

{

System.***out***.println(e);

}

**finally** {

System.***out***.println("........");

}

}

}

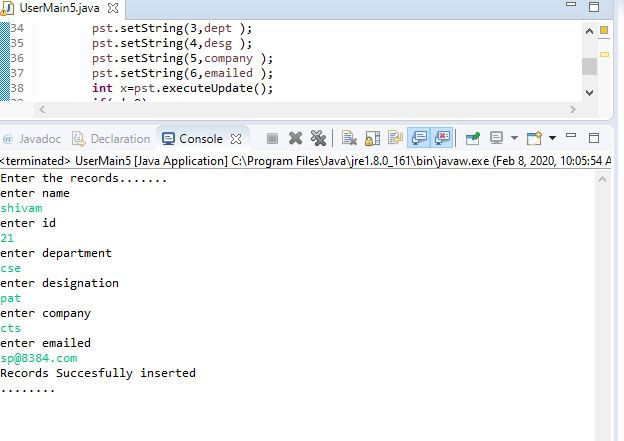
create database Employee;

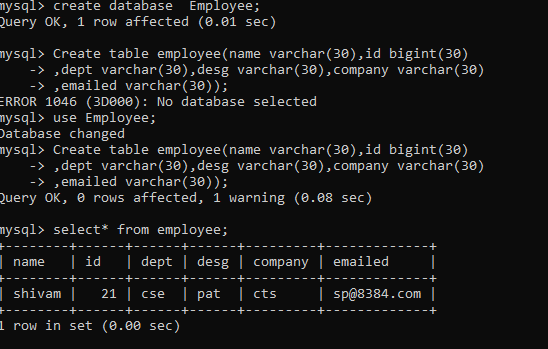
use databse Employee;

Create table employee(name varchar(30),id bigint(30)

,dept varchar(30),desg varchar(30),company varchar(30)

,emailed varchar(30));





4. Write a program that performs the following actions:

1. Read 2n integers as input & a set operator (of type char).

2. Create two arraylists to store n elements in each arraylist.

3. Write a function performSetOperations which accepts these two arraylist and the set operator as input.

4. The function would perform the following set operations:.

'+' for SET-UNION

'\*' for SET-INTERSECTION

'-' for SET-DIFFERENCE

5. Return the resultant arraylist.

Include a class UserMainCode with the static method performSetOperations which accepts two arraylist and returns an arraylist.

Create a Class Main which would be used to read 2n+1 integers and call the static method present in UserMainCode. (25 Marks)

**import** java.util.ArrayList;

**import** java.util.HashSet;

**import** java.util.List;

**import** java.util.Scanner;

**import** java.util.Set;

**public** **class** MainCode4 {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

ArrayList<Integer> ar1=**new** ArrayList<Integer>();

ArrayList<Integer> ar2=**new** ArrayList<Integer>();

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter the size of arraylist");

**int** size=sc.nextInt();

System.***out***.println("Enter elements in arraylist 1");

**for**(**int** i=0;i<size;i++)

{

ar1.add(sc.nextInt());

}

System.***out***.println("Enter elements in arraylist 2");

**for**(**int** i=0;i<size;i++)

{

ar2.add(sc.nextInt());

}

System.***out***.println("Enter operation u want to perform among +,-,\*");

String c=sc.next();

**char** ch=c.charAt(0);

System.***out***.println(ch);

List l1=*performSetOperations*(ar1,ar2,ch);

System.***out***.println(l1);

}

**private** **static** ArrayList performSetOperations(ArrayList<Integer> ar1, ArrayList<Integer> ar2, **char** ch) {

// **TODO** Auto-generated method stub

ArrayList l=**new** ArrayList();

Set set = **new** HashSet();

**switch**(ch) {

**case** '-':

**for** (**int** i = 0; i < ar1.size(); i++) {

**int** k = 0;

**for** (**int** j = 0; j < ar2.size(); j++) {

**if** (ar1.get(i) == ar2.get(j))

k = 1;

}

**if** (k == 0)

l.add(ar1.get(i));

}

**break**;

**case** '+':

set.addAll(ar1);

set.addAll(ar2);

**return** **new** ArrayList(set);

**case** '\*':

**for** (Integer t : ar1) {

**if**(ar2.contains(t)) {

l.add(t);

}

}

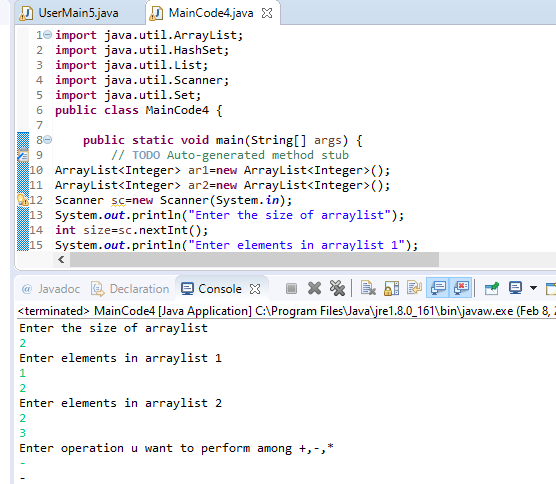
**return** l;

}

**return** l;

}

}



1. Write a program to read a string and validate PAN no. against following

rules: (15 Marks)

1. There must be eight characters.

2. First three letters must be alphabets followed by four digit number

and ends with alphabet

3. All alphabets should be in capital case.

**import** java.util.Scanner;

**class** User {

**public** **static** **int** validatePAN(String pan)

{

**int** r=0;

**if**(pan.length()==10)

{

**if**(pan.matches("[A-Z]{5}[0-9]{4}[A-Z]{1}"))

r=1;

**else**

r=2;

}

**return** r;

}

}

**public** **class** PanVerification

{

**public** **static** **void** main(String[] args)

{

System.***out***.println("Enter your PAN Number");

Scanner sc = **new** Scanner(System.***in***);

String pan=sc.next();

**int** r=User.*validatePAN*(pan);

**if**(r==1)

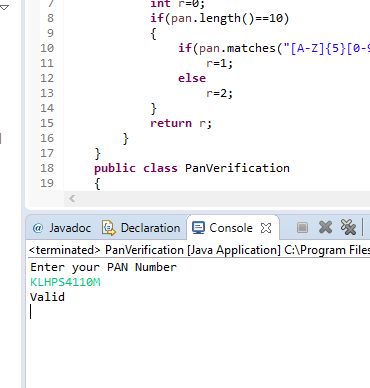
System.***out***.println("Valid");

**else**

System.***out***.println("Invalid");

}

}



2. Create user defined exceptions called InvalidAgeException and InsufficientAgeException, which has to be thrown when user enters a negative age and if user enters age less than 18 then it has to throw InSufficientAgeException , else has to show eligible for voting. (15 Marks)

**import** java.util.Scanner;

**class** InvalidAgeException **extends** Exception

{

**public** InvalidAgeException()

{

**super**("Age is Negative");

}

}

**class** InsufficientAgeException **extends** Exception

{

**public** InsufficientAgeException()

{

**super**("Age is Less Than 18");

}

}

**class** Voting

{

**public** **void** testAge(**int** age) **throws** InsufficientAgeException,InvalidAgeException

{

**if**(age>0 && age<18)

**throw** **new** InsufficientAgeException();

**else** **if**(age<0)

**throw** **new** InvalidAgeException();

**else**

System.***out***.println("Eligible for Voting");

}

}

**public** **class** ExceptionCase

{

**public** **static** **void** main(String[] args)

{

**int** age;

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter the age");

age=sc.nextInt();

Voting v=**new** Voting();

**try**

{

v.testAge(age);

}

**catch**(InsufficientAgeException e)

{

System.***out***.println(e);

}

**catch**(InvalidAgeException e)

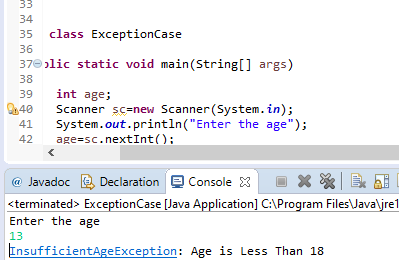
{

System.***out***.println(e);

}

}

}



3. Write a java program to read student objects in a ArrayList and display the student result based on the condition that max>=80, distinction, between 60 and 79 as First class, between 50 and 50 as Second class, else fail, and display only students who are passed. (25 Marks)

Note; Test the pass condition by using Java8 Predicate, find the class by using Function and display the results using Consumer predefined interfaces