

```
student student = (student) session.getSelection("student getByrollno", 103);
// print the student details
System.out.println(student.getrollno());
System.out.println(student.getName());
System.out.println(student.getMarks());
System.out.println(student.getPercentage());
```

## Assignment ~10.

1] JBCD - exception handling - string - array - collection  
static method, oops concept

- 1] Make a database 'db-bank'.
- 2] Make a table in it 'tbl-account'
- 3] Make a table in it 'tbl-transaction'  
(accountnumber, transaction date, amount,  
transaction type, balance)
- 4] Java program to add a details of the  
account holder method  
`saveDetails(account number, name, age,  
address, opening balance amount)`  
Transaction of bank account  
Method transaction(accountnumber, date,  
transaction type, amount)  
is deposit then add the amount and  
withdraw will minus the amount from  
the balance.

if balance is less than withdrawal then  
send exception 'Please check the amount!  
low balance'

```
package com.bank;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.math.BigDecimal;
import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
```

Integer accountNo;

// will load total-balance from tbl-account  
String getBalance = "select total-balance  
from db-bank.tbl-account where  
account-number = " + accountNo;

String getBalance = "select total-balance  
from db-banking.tbl-account where  
account-number = " + accountNo;  
// will insert record in table tbl-transaction

String creditTransactionQuery = "INSERT  
INTO 'db-bank'.'tbl-transaction' ('account  
-number', 'transactionDate', 'amount',  
'transaction-Type', 'balance') VALUES  
(?, ?, ?, ?, ?);";

```
String creditTransactionQuery = "INSERT  
INTO 'db-banking'.tbl-transaction ('acco-  
unt-number', 'transactionDate', 'amount',  
'transaction-Type', 'balance') VALUES (?, ?,  
?, ?, ?);
```

// will update the total balance after  
each transaction

```
String updateBalance = "UPDATE 'db-bank'  
• 'tbl-account' SET 'total-Balance' = "+  
totalBalance + "? WHERE 'account-number'  
= " + accountNo;
```

```
String updateBalance = "UPDATE 'db-bank-  
ing' • 'tbl-account' SET 'total-Balance' = "  
+ TotalBalance + "? WHERE 'account-number'  
= " + accountNo;
```

// method for getting JDBC connection

```
public void createConnection() {  
try {
```

```
    Class.forName("com.mysql.jdbc.Driver");  
    Connection con = DriverManager.getConnection  
    ("jdbc:mysql://localhost:3306/db-bank",  
    "root", "root");
```

// here db-bank is database name, root is  
username and password

```
    "jdbc:mysql://localhost:3306/db-banking",  
    "root", "root");
```

// here db-banking is database name, root  
is username and password.

```
    con.close();  
} catch(Exception e) {  
    @@ -46,10,+49,10 @@ public void saveCustomer(Integer acctNo, String name,  
    Integer age, String address
```

```
    try {  
        // load and establish connection to  
        JDBC driver
```

```
        Class.forName("com.mysql.jdbc.Driver");  
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3307/  
        db-bank", "root", "root");
```

```
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/db-banking",  
        "root", "root");  
        // query to insert record in tbl-account
```

```
        String query = "INSERT INTO 'db-bank'.  
        'tbl-account' ('account-number', 'name',  
        'age', 'address', 'opening-balance-  
        ammount', 'total-Balance') VALUES (?, ?,  
        ?, ?, ?, ?);";
```

```
        String query = "INSERT INTO 'db-banking'.  
        'tbl-account' ('account-number', 'name',  
        'age', 'address', 'opening-balance-  
        ammount', 'total-Balance') VALUES (?, ?,  
        ?, ?, ?, ?);";
```

```
PreparedStatement preparedstmt = con.  
    prepareStatement(query);  
    // load values from constructor to table  
    // columns
```

```
public void saveCustomer(Integer acctNo,  
    String name, Integer age, String address
```

```
System.out.println("The user: " + name  
    + " with Account No: " + acctNo  
    + " has been added successfully. !");
```

```
// as account created its default first  
// transaction to credit opening amount  
// in account
```

```
String query2 = "INSERT INTO 'db-bank'.  
    'tbl-transaction'('account-number', 'tra-  
    nsactionDate', 'amount', 'transaction-  
    Type', 'balance') VALUES (?, ?, ?, ?, ?);
```

```
String query2 = "INSERT INTO 'db-banking'  
    . 'tbl-transaction'('account-number', 'tra-  
    nsactionDate', 'amount', 'transaction-  
    Type', 'balance') VALUES (?, ?, ?, ?, ?);
```

```
PreparedStatement preparedstmt2 = con.  
    prepareStatement(query2);
```

```
// load values from constructor to table  
// columns
```

```
public void creditTransaction(Integer
```

```
acctNo, BigDecimal ammount){  
    // load and establish connection to  
    // JDBC driver
```

```
Class.forName("com.mysql.jdbc.Driver");  
Connection con = DriverManager.getConnection  
("jdbc:mysql://localhost:3307/db-bank",  
"root", "root");
```

```
Connection con = DriverManager.getConnection  
("jdbc:mysql://localhost:3306/db-banking"  
, "root", "root");  
// load balance ammount before actual  
transaction
```

```
Statement stmt = con.createStatement();  
// query to get total - balance from  
tbl-account
```

```
String getBalance = "select total - balance  
from db-bank .tbl-account where account-  
number = " + acctNo;
```

```
String getBalance = "select total - balance  
from db-banking .tbl - account where  
account - number = " + acctNo;
```

```
ResultSet rs = stmt.executeQuery(getBalance);  
BigDecimal totalBalance = null;  
while(rs.next())
```

```
public void creditTransaction(Integer acctNo,  
        BigDecimal ammount) {
```

```
System.out.println("Credit transaction");
// will update total amount after
credit operation
```

```
String updateBalance = "UPDATE 'db-bank'.
    'tbl-account' SET 'total-Balance' = ,
    + totalBalance + " WHERE 'account-number'
    = " + acctNo;
```

```
String updateBalance = " UPDATE 'db-bank-
ing', 'tbl-account' SET 'total-Balance'
= " + totalBalance + " WHERE 'account-num-
ber' = " + acctNo;
```

```
String updateBalance = " UPDATE 'db-banking'
    . 'tbl-account' SET 'total-Balance' = "
    + totalBalance + " WHERE 'account-number'
    = " + acctNo;
```

```
p PreparedStatement preparedstmt = con.
    preparedStatement(updateBalance);
System.out.println("balance updated");
preparedstmt.execute();
```

```
public void withdrawTransaction(Integer
    acctNo, BigDecimal ammount) {
    // load and establish connection to JDBC
    driver
```

```
(Class.forName("com.mysql.jdbc.Driver"));
```

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3307/db-bank", "root", "root");
```

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/db-banking", "root", "root");
```

// load balance ammount before actual transaction

```
Statement stmt = con.createStatement();
```

// query to get total -balance from  
tbl-account

```
String getBalance = "select total -balance  
from db-bank .tbl-account where  
account-number = " + acctNo;
```

```
String getBalance = "select total -balance  
from db-banking .tbl-account where  
account-number = " + acctNo;
```

```
ResultSet rs = stmt.executeQuery(getBalance);  
BigDecimal totalBalance = null;
```

```
public void withdrawTransaction(Integer  
acctNo, BigDecimal ammount) {
```

```
System.out.println("Credit transaction");  
// Update balance in account after withdraw
```

```
String updateBalance = "UPDATE 'db-bank',  
'tbl-account' SET 'total-Balance' = ''  
+ totalBalance + " WHERE 'account-number'  
= " + acctNo;
```

```
String updateBalance = "UPDATE 'db-ban-  
king', 'tbl-account' SET 'total-Balance'  
= " + totalBalance + " WHERE 'account-  
number' = " + acctNo;
```

```
PreparedStatement preparedstmt = con.  
prepareStatement(updateBalance);
```

```
// print total balance  
public void printPassBook(Integer acctNo){  
try {  
// load and establish connection to JDBC  
driver
```

```
Class.forName("com.mysql.jdbc.Driver");  
Connection con = DriverManager.getConnection  
("jdbc:mysql://localhost:3307/db-bank",  
"root", "root");
```

```
Connection con = DriverManager.getConnection  
("jdbc:mysql://localhost:3306/db-banking",  
"root", "root");
```

```
// select rows from both tbl-account and  
tbl-transaction tables
```

```
Statement stmt = con.createStatement();
```

```
String getAllTransactions = "select transaction-  
onDate, amount, transaction-Type, balance  
FROM db-bank.tbl-transaction where  
account-number = " + acctNo;
```

```
String getAcctDetails = "select * from  
db-bank.tbl-account where account-  
number = " + acctNo;
```

```
String getAllTransactions = "select transaction-  
onDate, amount, transaction-Type,  
balance FROM db-banking.tbl-transaction  
where account-number = " + acctNo;
```

```
String getAcctDetails = "select * from db-  
banking.tbl-account where account-  
number = " + acctNo;
```

```
ResultSet rs = stmt.executeQuery(getAcct-  
Details);
```

```
public void printPassBook(Integer acctNo){  
    // create file with User's name
```

```
String tileName = name + ".txt";  
File tileObj = new File(tileName);  
if(tileObj.createNewFile()) {  
    System.out.println("File created : " +  
        tileObj.getName());  
}
```

```
// File tileobj = new File(tileName);  
// if(tileobj.createNewFile()) {  
//     System.out.println("File created : " +  
//         tileObj.getName());  
}
```

// }

// create FileWriter object to write data  
in file

FileWriter myWriter = new FileWriter(tileName);

PrintWriter outputfile = null;

outputfile = new PrintWriter(tileName);

// String Buffer to load into to file.

StringBuffer str = new StringBuffer();

// Add account info

str.append("Account Number:").append(acctNo);

str.append("Name:").append(name);

str.append("Age:").append(age);

str.append("Address:").append(address);

str.append("\n\nDATE").append(date);

str.append("Transaction").append(transaction);

str.append("Amount").append(amount);

str.append("Balance").append(balance);

myWriter.write(str.toString());

outputfile.append(str.toString());

rs = stmt.executeQuery(getAllTransactions);

while(rs.next()) {

StringBuffer str1 = new StringBuffer();

StringBuilder str2 = new StringBuilder();

// add transaction info

str1.append(rs.getDate("transactionDate"))

str1.append(")").append(rs.getBigDecimal("amount"))

str1.append(")").append(rs.getString("transaction-Type"))

str1.append(")").append(rs.getBigDecimal("balance"));

str1.append(rs.getBigDecimal("balance"));

```
myWriter.write(str1.toString());
outputfile.append("\n" + str1.toString());
}
```

```
// Shows msg in screen that PassBook  
is printed intile
```

```
System.out.println("PassBook printed  
successfully... !!");
outputfile.close();
```

```
? catch(Exception e) { // raise the exception if
```

```
? catch(FileNotFoundException | ClassNotFoundException | SQLException e) { // raise  
any  
the exception if any
? }
```

```
public void deleteAccount(Integer acctNo){  
try {
```

```
Class.forName("com.mysql.jdbc.Driver");
Connection con = DriverManager.getConnection
("jdbc:mysql://localhost:3307/db-
bank", "root");
```

```
String sqlacc = "SELECT name FROM 'tbl-
account' WHERE acctno = " + no + " , , ";
ResultSet rs = connection.createStatement().  
executeQuery(sqlacc);
```

```
String name = null;
while(rs.next()) {
    name = rs.getString("name");
}
```

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306  
lab-banking", "root", "root");
```

```
String getAcctDetails = "select * from  
db-banking.tbl-account where account  
number = " + acctNo;
```

```
Statement stmt = con.createStatement();  
ResultSet rs = stmt.executeQuery(get  
AcctDetails);
```

```
String name = "demo", address = "";  
Integer age = 0;
```

```
BigDecimal openingBalance = null, totalB  
alance = null;
```

```
while (rs.next()) {  
// Load table data into variables
```

```
name = rs.getString(2);
```

```
address = rs.getString(4);
```

```
age = rs.getInt(3);
```

```
openingBalance = rs.getBigDecimal(5);
```

```
totalBalance = rs.getBigDecimal(6);
```

```
}
```

```
String filename = name + ".txt";
```

```
System.out.println("filename" + filename);
```

```
File f = new File(filename); // file to be  
it (f.delete()) delete
```

```
// not running, will be removed
```



