Assignment-1 A **Prashanth S**

19MID0020

Question

Assessment 1A

Mellon share owner services maintain the details of the share holders.

a) Maintain the personal details of the share holder.

b) Manipulate the users to buy and sell the shares.

c) Maintain the account balances after sell and buying the shares.

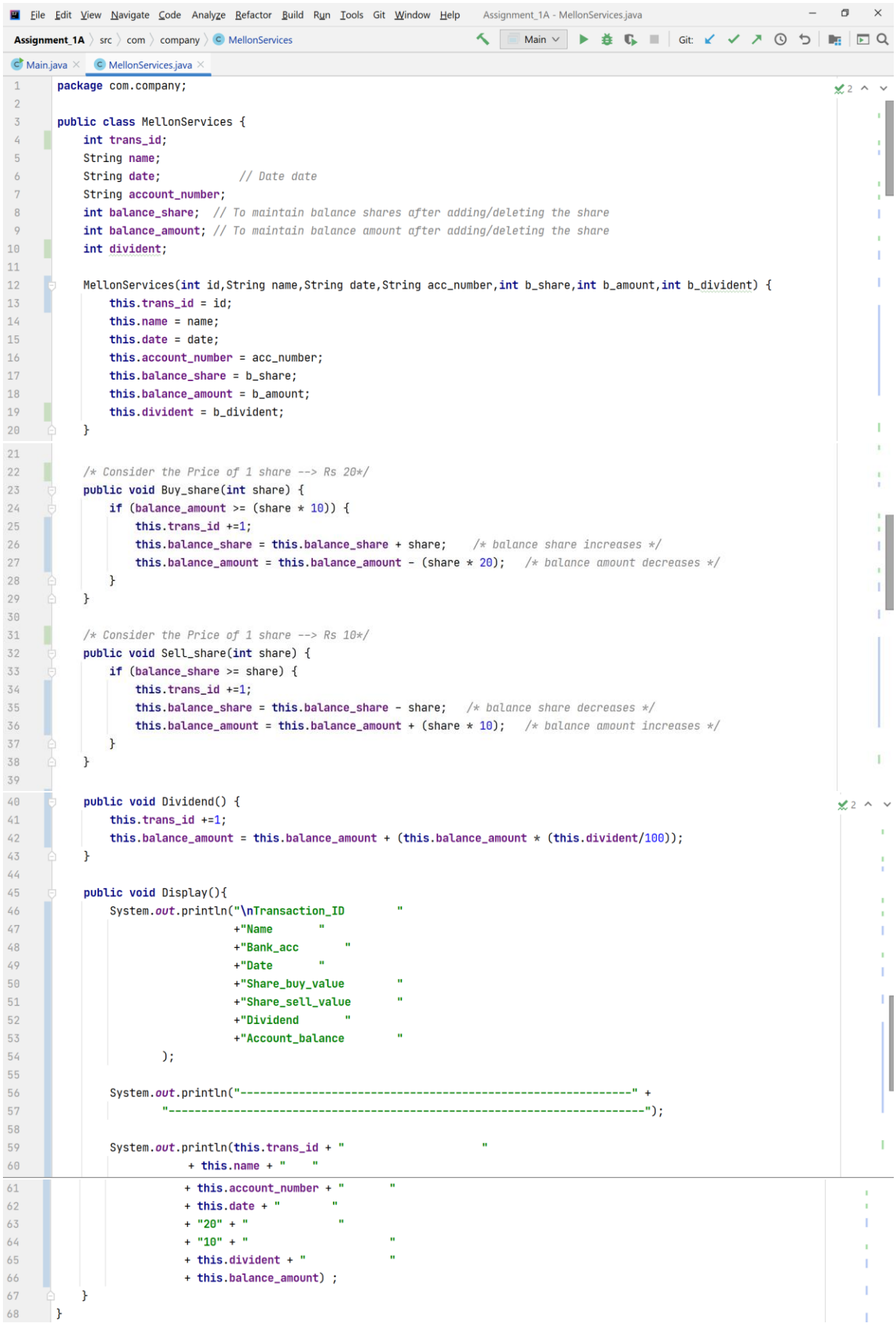
d) Issue the dividends yearly once. Dividends means a bonus. E.g. 30% bonus means balance = balance + (balance * 0.3)

Write a Java program to implement the problem stated above using classes and objects in Java

Expected output:

.Tran Id	Share holder name	Bank Acc No	Date	Share Sell Value	Shares buy Value	Dividend	Acc Bal

Melon Class




```java
package com.company;

public class MellonServices {
    int trans_id;
    String name;
    String date;            // Date date
    String account_number;
    int balance_share;  // To maintain balance shares after adding/deleting the share
    int balance_amount; // To maintain balance amount after adding/deleting the share
    int divident;

    MellonServices(int id,String name,String date,String acc_number,int b_share,int b_amount,int b_divident) {
        this.trans_id = id;
        this.name = name;
        this.date = date;
        this.account_number = acc_number;
        this.balance_share = b_share;
        this.balance_amount = b_amount;
        this.divident = b_divident;
    }

    /* Consider the Price of 1 share --> Rs 20*/
    public void Buy_share(int share) {
        if (balance_amount >= (share * 10)) {
            this.trans_id +=1;
            this.balance_share = this.balance_share + share;    /* balance share increases */
            this.balance_amount = this.balance_amount - (share * 20);   /* balance amount decreases */
        }
    }

    /* Consider the Price of 1 share --> Rs 10*/
    public void Sell_share(int share) {
        if (balance_share >= share) {
            this.trans_id +=1;
            this.balance_share = this.balance_share - share;   /* balance share decreases */
            this.balance_amount = this.balance_amount + (share * 10);   /* balance amount increases */
        }
    }

    public void Dividend() {
        this.trans_id +=1;
        this.balance_amount = this.balance_amount + (this.balance_amount * (this.divident/100));
    }

    public void Display(){
        System.out.println("\nTransaction_ID        "
                           +"Name       "
                           +"Bank_acc       "
                           +"Date       "
                           +"Share_buy_value        "
                           +"Share_sell_value       "
                           +"Dividend       "
                           +"Account_balance        "
                   );

        System.out.println("-------------------------------------------------------------" +
                "-------------------------------------------------------------------------");

        System.out.println(this.trans_id + "                    "
                    + this.name + "    "
                    + this.account_number + "       "
                    + this.date + "        "
                    + "20" + "              "
                    + "10" + "                      "
                    + this.divident + "             "
                    + this.balance_amount) ;
    }
}
```

Main method

File Edit View Navigate Code Analyze Refactor Build Run Tools Git Window Help — Assignment_1A - Main.java

Assignment_1A 〉 src 〉 com 〉 company 〉 Main 〉 main — Main — Git:

Main.java | MellonServices.java

```java
package com.company;

public class Main {

    public static void main(String[] args) {
        MellonServices obj1 = new MellonServices( id: 1, name: "Prashanth", date: "2020-02-08", acc_number: "2ABC123", b_share: 25, b_am
        MellonServices obj2 = new MellonServices( id: 2, name: "William", date: "2020-11-23", acc_number: "9PQR456", b_share: 30, b_amou
        MellonServices obj3 = new MellonServices( id: 3, name: "Monica", date: "2019-04-29", acc_number: "8XYZ789", b_share: 40, b_amount

        System.out.println("      ##################           Before transactions           ####################");
        obj1.Display();
        obj2.Display();
        obj3.Display();

        obj1.Buy_share(30);  // Prashanth wants to buy 10 shares
        obj1.Buy_share(20);  // William wants to buy 30 shares
        obj1.Buy_share(10);  // Monica wants to buy 40 shares

        obj1.Sell_share(5);  // Prashanth wants to sell 5 shares
        obj2.Sell_share(10); // William wants to sell 10 shares
        obj3.Sell_share(15); // Monica wants to sell 15 shares

        obj1.Dividend();        // Prashanth wants to give divident as  5%
        obj1.Dividend();        // William wants to give divident as  10%
        obj1.Dividend();        // Monica wants to give divident as  15%

        System.out.println();
        System.out.println("      ##################           After transactions           ####################");
        obj1.Display();
        obj2.Display();
        obj3.Display();
    }
```

Output

```
"C:\Program Files\Amazon Corretto\jdk15.0.2_7\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edit
      ##################           Before transactions           ####################

Transaction_ID      Name      Bank_acc      Date      Share_buy_value      Share_sell_value      Dividend      Account_
-----------------------------------------------------------------------------------------------------------------------
1                   Prashanth    2ABC123      2020-02-08       20             10                   5            1000

Transaction_ID      Name      Bank_acc      Date      Share_buy_value      Share_sell_value      Dividend      Account_
-----------------------------------------------------------------------------------------------------------------------
2                   William    9PQR456      2020-11-23       20            10                   10           1500

Transaction_ID      Name      Bank_acc      Date      Share_buy_value      Share_sell_value      Dividend      Account_
-----------------------------------------------------------------------------------------------------------------------
3                   Monica    8XYZ789      2019-04-29       20            10                   15           2000

      ##################           After transactions           ####################

Transaction_ID      Name      Bank_acc      Date      Share_buy_value      Share_sell_value      Dividend      Account_
-----------------------------------------------------------------------------------------------------------------------
7                   Prashanth    2ABC123      2020-02-08       20             10                   5            50

Transaction_ID      Name      Bank_acc      Date      Share_buy_value      Share_sell_value      Dividend      Account_
-----------------------------------------------------------------------------------------------------------------------
3                   William    9PQR456      2020-11-23       20            10                   10           1600

Transaction_ID      Name      Bank_acc      Date      Share_buy_value      Share_sell_value      Dividend      Account_
-----------------------------------------------------------------------------------------------------------------------
4                   Monica    8XYZ789      2019-04-29       20            10                   15           2150

Process finished with exit code 0
```

Melon Class

```
package com.company;

public class MellonServices {
    int trans_id;
    String name;
    String date;             // Date date
    String account_number;
    int balance_share;  // To maintain balance shares after
adding/deleting the share
    int balance_amount; // To maintain balance amount after
adding/deleting the share
    int divident;

    MellonServices(int id,String name,String date,String
acc_number,int b_share,int b_amount,int b_divident) {
        this.trans_id = id;
        this.name = name;
        this.date = date;
        this.account_number = acc_number;
        this.balance_share = b_share;
        this.balance_amount = b_amount;
        this.divident = b_divident;
    }

    /* Consider the Price of 1 share --> Rs 20*/
    public void Buy_share(int share) {
        if (balance_amount >= (share * 10)) {
            this.trans_id +=1;
            this.balance_share = this.balance_share + share;
/* balance share increases */
            this.balance_amount = this.balance_amount - (share
* 20);   /* balance amount decreases */
        }
    }

    /* Consider the Price of 1 share --> Rs 10*/
    public void Sell_share(int share) {
        if (balance_share >= share) {
            this.trans_id +=1;
            this.balance_share = this.balance_share - share;
/* balance share decreases */
            this.balance_amount = this.balance_amount + (share
* 10);   /* balance amount increases */
        }
    }

    public void Dividend() {
        this.trans_id +=1;
        this.balance_amount = this.balance_amount +
```

```
(this.balance_amount * (this.divident/100));
    }

    public void Display(){
        System.out.println("\nTransaction_ID        "
                           +"Name        "
                           +"Bank_acc        "
                           +"Date        "
                           +"Share_buy_value        "
                           +"Share_sell_value        "
                           +"Dividend        "
                           +"Account_balance        "
                );

        System.out.println("---------------------------------
-------------------------" +
                "-----------------------------------------------
----------------------------");

        System.out.println(this.trans_id + "
"
                    + this.name + "    "
                    + this.account_number + "        "
                    + this.date + "         "
                    + "20" + "                "
                    + "10" + "                        "
                    + this.divident + "                "
                    + this.balance_amount) ;
    }
}
```

Main method

```
package com.company;

public class Main {

    public static void main(String[] args) {
        MellonServices obj1 = new
MellonServices(1,"Prashanth", "2020-02-08", "2ABC123", 25,
1000,5);
        MellonServices obj2 = new MellonServices(2,"William",
"2020-11-23", "9PQR456", 30, 1500,10);
        MellonServices obj3 = new MellonServices(3,"Monica",
"2019-04-29", "8XYZ789", 40, 2000,15);

        System.out.println("      ##################
Before transactions             ####################");
        obj1.Display();
        obj2.Display();
        obj3.Display();
```

```
        obj1.Buy_share(30);  // Prashanth wants to buy 10
shares
        obj1.Buy_share(20);  // William wants to buy 30 shares
        obj1.Buy_share(10);  // Monica wants to buy 40 shares

        obj1.Sell_share(5);  // Prashanth wants to sell 5
shares
        obj2.Sell_share(10); // William wants to sell 10
shares
        obj3.Sell_share(15); // Monica wants to sell 15 shares

        obj1.Dividend();        // Prashanth wants to give
divident as  5%
        obj1.Dividend();        // William wants to give
divident as  10%
        obj1.Dividend();        // Monica wants to give
divident as  15%

        System.out.println();
        System.out.println("      ##################
After transactions             ####################");
        obj1.Display();
        obj2.Display();
        obj3.Display();
    }
}
```

Output

```
"C:\Program Files\Amazon Corretto\jdk15.0.2_7\bin\java.exe" "-
javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community
Edition 2020.3.2\lib\idea_rt.jar=50717:C:\Program
Files\JetBrains\IntelliJ IDEA Community Edition 2020.3.2\bin"
-Dfile.encoding=UTF-8 -classpath "C:\WinterSemester-
2021\CSI2008 Programming in JAVA\JAVA lab
practice\Assignment_1A\out\production\Assignment_1A"
com.company.Main
      ##################           Before transactions
####################

Transaction_ID        Name       Bank_acc       Date
Share_buy_value        Share_sell_value       Dividend
Account_balance
-----------------------------------------------------------------
-----------------------------------------------------------------
---------
1                     Prashanth    2ABC123       2020-02-08
20              10                        5              1000
```

```
Transaction_ID        Name       Bank_acc       Date
Share_buy_value        Share_sell_value       Dividend
Account_balance
-------------------------------------------------------------
-------------------------------------------------------------
---------
2                     William    9PQR456       2020-11-23
20              10                        10             1500

Transaction_ID        Name       Bank_acc       Date
Share_buy_value        Share_sell_value       Dividend
Account_balance
-------------------------------------------------------------
-------------------------------------------------------------
---------
3                     Monica    8XYZ789       2019-04-29
20              10                        15             2000

      ##################            After transactions
####################

Transaction_ID        Name       Bank_acc       Date
Share_buy_value        Share_sell_value       Dividend
Account_balance
-------------------------------------------------------------
-------------------------------------------------------------
---------
7                     Prashanth    2ABC123       2020-02-08
20              10                        5              50

Transaction_ID        Name       Bank_acc       Date
Share_buy_value        Share_sell_value       Dividend
Account_balance
-------------------------------------------------------------
-------------------------------------------------------------
---------
3                     William    9PQR456       2020-11-23
20              10                        10             1600

Transaction_ID        Name       Bank_acc       Date
Share_buy_value        Share_sell_value       Dividend
Account_balance
-------------------------------------------------------------
-------------------------------------------------------------
---------
4                     Monica    8XYZ789       2019-04-29
20              10                        15             2150

Process finished with exit code 0
```

Github : https://github.com/PrashanthSingaravelan/WinterSemester-2021/tree/main/CSI2008%20Programming%20in%20JAVA/JAVA%20lab%20practice/Assignment_1A_1B

Assessment 1B

Hyundai car show room sells the cars in weekly basis. It has a sales module to perform the sale of a car. The functionality of the sales module are

a. Constructor functions performs the input activity like getting the details of car(Model, Color, Engine Type, Fuel Type, Price)
b. Authorize function check the payment (Should be greater than 50 percent of unit price) made by the user and approves the sale order and calls the other function called order processing.
c. Order processing function check the stock for availability of the car and prints the delivery order(format given below) or other wise it will prints the production order (format given below).

Delivery Order:

Owner Name	Car Model	Color	Fuel Type	Delivery date

Production Order:

Production branch Name	Showroom address	Car Model	Color	Engine type	Date of delivery

Write a Java program to implement the concept of constructors for the

problem stated above. Copy constructor, default constructor, multiple argument constructors need to be there.

Data-base creation using Python

In [1]:
```
import sqlite3
```

In [2]:
```
db = sqlite3.connect("cars.db")
```

In [3]:
```
c = db.cursor()
```

In [4]:
```
## Creating the hyundai table
c.execute('CREATE TABLE hyundai(model text,color text,engine_type text ,fuel_type text,unit_price int)')
```

Out[4]:
```
<sqlite3.Cursor at 0x239dc57e490>
```

In [6]:
```
## Inserting the data in-to the table
c.execute("INSERT INTO hyundai VALUES ('baleno', 'blue', 'Automatic', 'petrol',700000)")
c.execute("INSERT INTO hyundai VALUES ('ciaz', 'red', 'Automated_Manual_Transmission', 'petrol', 900000)")
c.execute("INSERT INTO hyundai VALUES ('s-cross', 'black', 'Manual', 'petrol', 1100000)")
c.execute("INSERT INTO hyundai VALUES ('xl-6', 'red', 'Manual', 'petrol', 1000000)")
c.execute("INSERT INTO hyundai VALUES ('swift', 'white', 'Automatic', 'diesel', 500000)")
```

Out[6]:
```
<sqlite3.Cursor at 0x239dc57e490>
```

In [9]:
```
data = c.execute("SELECT * FROM hyundai")
for i in data:
    print(i)
```

```
('baleno', 'blue', 'Automatic', 'petrol', 700000)
('ciaz', 'red', 'Automated_Manual_Transmission', 'petrol', 900000)
('s-cross', 'black', 'Manual', 'petrol', 1100000)
('xl-6', 'red', 'Manual', 'petrol', 1000000)
('swift', 'white', 'Automatic', 'diesel', 500000)
```

These data-base can be created using JAVA, so I created the data-base and inserted values using JDBC.

Hyundai class

Structure of the Hyundai class

```
1   package com.company;
2   import java.sql.*;
3   import java.util.Calendar;
4
5   public class Hyundai {
6       String model;         /* baleno, ciaz, s-cross, xl-6 */
7       String color;         /* red, blue, white, black */
8       String engine_type;   /* Automatic, Automated Manual Transmission(AMT), Manual */
9       String fuel_type;     /* petrol, diesel */
10      int unit_price;       /* price of the car from the factory */
11      int payment;         /* payment made by the customer */
12
13                           /* Paramterized Constructor */
14      Hyundai(String data) {...}
40                         /* To display all the car details */
41      public void Display() {...}
48
49                           /* Paramterized Constructor */
50      Hyundai(String user_model,String user_color, String user_engine_type,String user_fuel_type,int user_unit_price) {...}
57
58      public void Authorize(Hyundai[] obj1) {...}
72
73      public void Order_Processing(Hyundai[] obj1) {...}
96
97      public void Deliver_Order() {...}
109
110     public void Production_Order() {...}
```

Hyundai class

```java
package com.company;
import java.sql.*;
import java.util.Calendar;

public class Hyundai {
    String model;        /* baleno, ciaz, s-cross, xl-6 */
    String color;        /* red, blue, white, black */
    String engine_type;  /* Automatic, Automated Manual Transmission(AMT), Manual */
    String fuel_type;    /* petrol, diesel */
    int unit_price;      /* price of the car from the factory */
    int payment;        /* payment made by the customer */

                        /* Paramterized Constructor */
    Hyundai(String data) {
        try {
            Connection con = DriverManager.getConnection    /*creating the connection*/
            ( url: "jdbc:sqlite:C://WinterSemester-2021//CSI2008 Programming in JAVA//JAVA lab practice//Assignment_1A_1B//car

            Statement stm = con.createStatement(); // creating the statem
            stm.execute( sql: "SELECT * FROM nexa WHERE model ='" + data + "'");
            ResultSet result = stm.getResultSet();   // output of the query is stored in result

            while (result.next()) {
                this.model = result.getString( columnLabel: "model");
                this.color = result.getString( columnLabel: "color");
                this.engine_type = result.getString( columnLabel: "engine_type");
                this.fuel_type   = result.getString( columnLabel: "fuel_type");
                this.unit_price  = result.getInt( columnLabel: "unit_price");
            }
            result.close();

            con.setAutoCommit(true);
            stm.close();
            con.close();
        }
        catch (SQLException e) {
            System.out.println("Some-thing went wrong " + e.getMessage());
        }
    }
                /* To display all the car details */
    public void Display() {...}

                    /* Paramterized Constructor */
    Hyundai(String user_model,String user_color, String user_engine_type,String user_fuel_type,int user_unit_price) {
        this.model = user_model;
        this.color = user_color;
        this.engine_type = user_engine_type;
        this.fuel_type   = user_fuel_type;
        this.unit_price = user_unit_price;
```

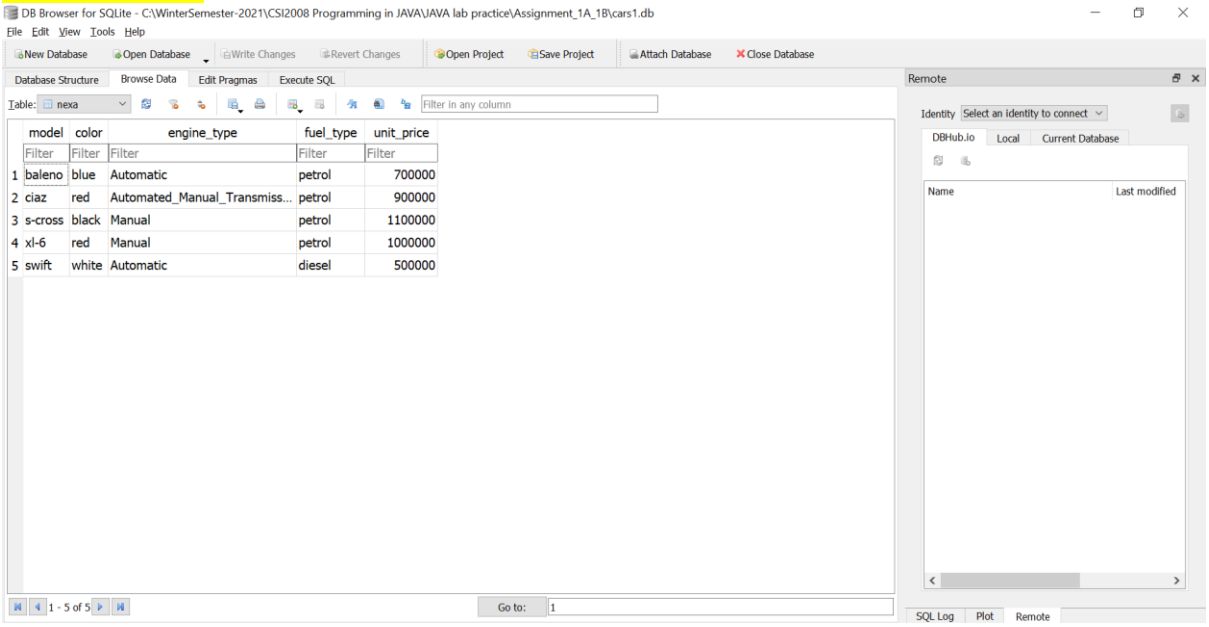
```java
    }

    public void Authorize(Hyundai[] obj1) {
        int flag = 0;
        for (int i = 0; i < 5; i++) {
            if (this.unit_price == (obj1[i].unit_price/2)) {
                flag = 1;
            }
        }
        if (flag==1) {
            this.Order_Processing(obj1);
        }
        else if (flag==0){
            System.out.println("Customer, please make your half payment");
        }
    }

    public void Order_Processing(Hyundai[] obj1) {
        System.out.println("Order processing is going on");
        int flag = 0;
            /*Checking the availability of stocks*/
        for (int i=0;i<5;i++) {
            if (
                    (this.model.equals(obj1[i].model)) &&
                    (this.color.equals(obj1[i].color)) &&
                    (this.engine_type.equals(obj1[i].engine_type)) &&
                    (this.fuel_type.equals(obj1[i].fuel_type))
                    /*(this.unit_price >= (obj1[i].unit_price)/2)*/
            ) {
                flag = 1;
            }
        }

            if (flag==1) {
                this.Deliver_Order();
            }
            else {
                this.Production_Order();
            }
    }

    public void Deliver_Order() {
        System.out.println("//////////////////////////////////////////////////////////////////////////////");
        System.out.println("Delivery order is sent to the customer");
        System.out.println("Owner name : Prashanth");
        System.out.println("Car model : " + this.model);
        System.out.println("Color : " + this.color);
        System.out.println("Fuel type : " + this.fuel_type);

        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.DATE, amount: +10);
        System.out.println("Delivery date from 10 days of purchase : " + cal.getTime());
    }

    public void Production_Order() {
        System.out.println("//////////////////////////////////////////////////////////////////////////////");
        System.out.println("Production order is sent to the factory");

        System.out.println("Production branch name : Avadi");
        System.out.println("Showroom address : No:47,Avadi,Chennai-54");
        System.out.println("Car model : " + this.model);
        System.out.println("Color : " + this.color);
        System.out.println("Engine type : " + this.engine_type);

        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.DATE, amount: +20);
        System.out.println("Delivery date from 20 days of purchase : " + cal.getTime());
    }
}
```

Main function

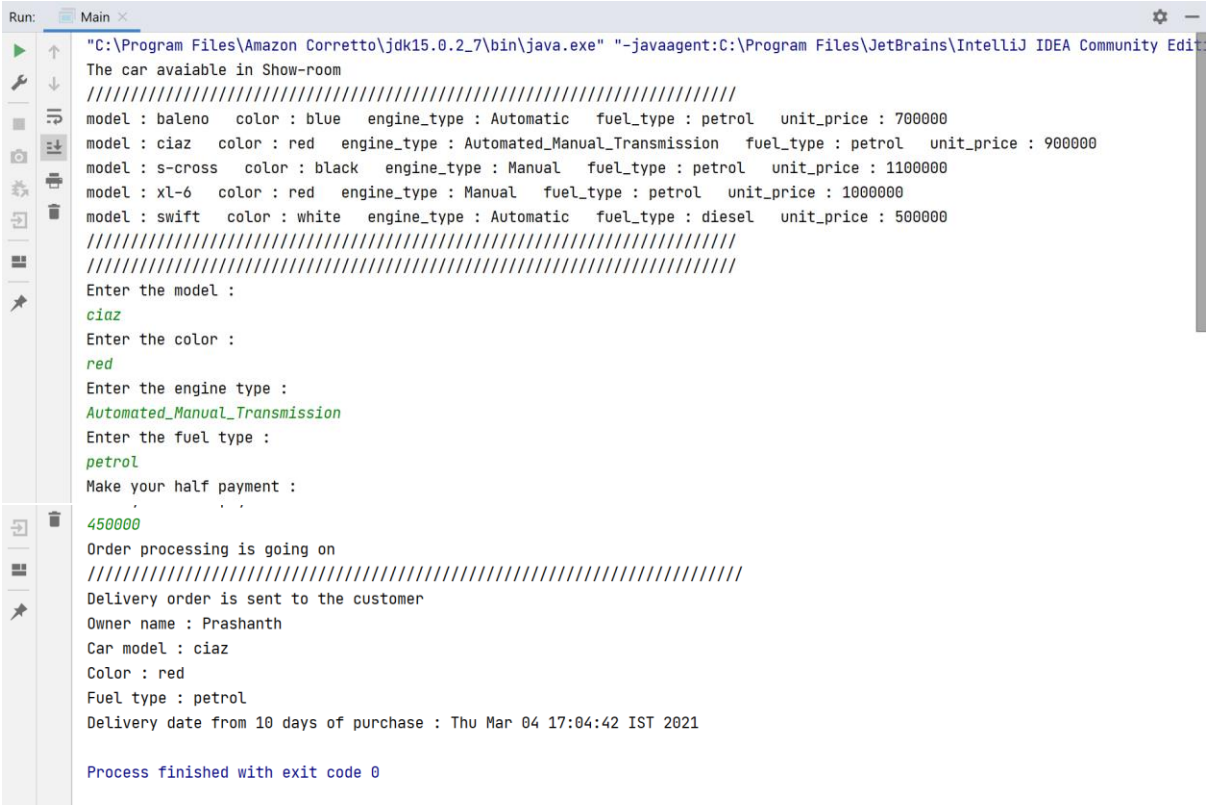
```java
package com.company;

import java.sql.*;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        try {
            /*  Data-base creation  */
            Connection con = DriverManager.getConnection    /*creating the connection*/
( url: "jdbc:sqlite:C://WinterSemester-2021//CSI2008 Programming in JAVA//JAVA lab practice//Assignment_1A_1B//cars1.d

            Statement stm = con.createStatement(); // creating the statement object

            stm.execute(
                    sql: "CREATE TABLE IF NOT EXISTS nexa" +
                            "(model text,color text,engine_type text , fuel_type text,unit_price int)");

            stm.execute( sql: "INSERT INTO nexa VALUES ('baleno', 'blue', 'Automatic', 'petrol',700000)");
            stm.execute( sql: "INSERT INTO nexa VALUES ('ciaz', 'red', 'Automated_Manual_Transmission', 'petrol', 9000
            stm.execute( sql: "INSERT INTO nexa VALUES ('s-cross', 'black', 'Manual', 'petrol', 1100000)");
            stm.execute( sql: "INSERT INTO nexa VALUES ('xl-6', 'red', 'Manual', 'petrol', 1000000)");
            stm.execute( sql: "INSERT INTO nexa VALUES ('swift', 'white', 'Automatic', 'diesel', 500000)");

            con.setAutoCommit(true);
            stm.close();
            con.close();
        }

        catch (SQLException e) {
            System.out.println("Some-thing went wrong " + e.getMessage());
        }

        Hyundai[] obj1;
        obj1 = new Hyundai[5];

        obj1[0] = new Hyundai( data: "baleno");
        obj1[1] = new Hyundai( data: "ciaz");
        obj1[2] = new Hyundai( data: "s-cross");
        obj1[3] = new Hyundai( data: "xl-6");
        obj1[4] = new Hyundai( data: "swift");

        /* Showing the available cars in the showroom to the customer */
        System.out.println("The car avaiable in Show-room");
        System.out.println("////////////////////////////////////////////////////////////////////////////");
        for(int i=0;i<5;i++)  {  obj1[i].Display(); }
        System.out.println("////////////////////////////////////////////////////////////////////////////");

        /* Getting input from the customer */
        Scanner input = new Scanner(System.in);
        String user_model,user_color,user_engine_type,user_fuel_type;
        int user_unit_price;
```

```
53
54          System.out.println("////////////////////////////////////////////////////////////////////////////");
55          System.out.println("Enter the model : ");
56          user_model = input.nextLine();
57
58          System.out.println("Enter the color : ");
59          user_color = input.nextLine();
60
61          System.out.println("Enter the engine type : ");
62          user_engine_type = input.nextLine();
63
64          System.out.println("Enter the fuel type : ");
65          user_fuel_type = input.nextLine();
66
67          System.out.println("Make your half payment : ");
68          user_unit_price = input.nextInt();
69
70          /* Creating a user object */
71          Hyundai user = new Hyundai(user_model,user_color,user_engine_type,user_fuel_type,user_unit_price);
72          user.Authorize(obj1);
73
74      }
75
```

Data-base

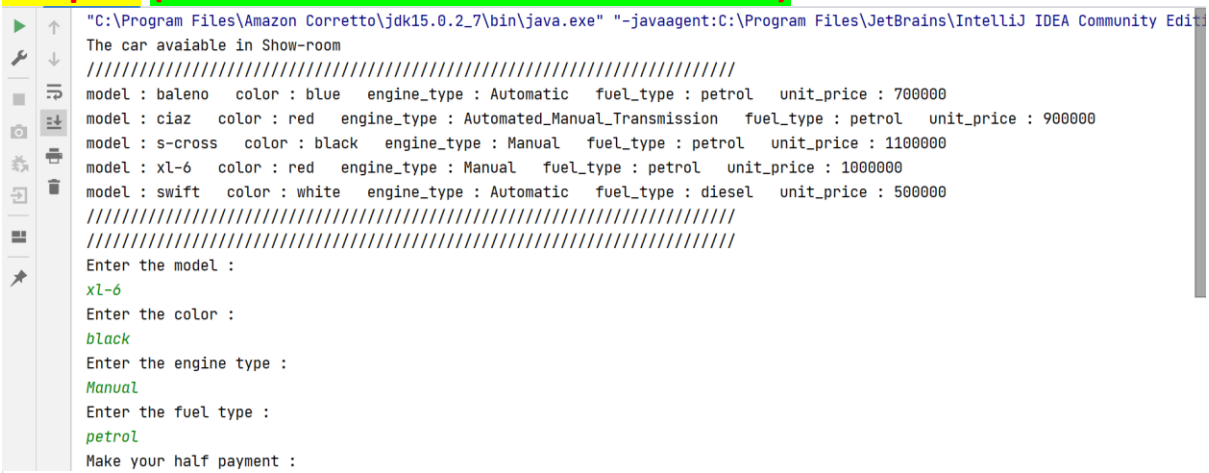


Output: (User-choice available in Stocks)



```
Run:  Main
"C:\Program Files\Amazon Corretto\jdk15.0.2_7\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edit
The car avaiable in Show-room
////////////////////////////////////////////////////////////////////////////
model : baleno   color : blue   engine_type : Automatic   fuel_type : petrol   unit_price : 700000
model : ciaz   color : red   engine_type : Automated_Manual_Transmission   fuel_type : petrol   unit_price : 900000
model : s-cross   color : black   engine_type : Manual   fuel_type : petrol   unit_price : 1100000
model : xl-6   color : red   engine_type : Manual   fuel_type : petrol   unit_price : 1000000
model : swift   color : white   engine_type : Automatic   fuel_type : diesel   unit_price : 500000
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
Enter the model :
ciaz
Enter the color :
red
Enter the engine type :
Automated_Manual_Transmission
Enter the fuel type :
petrol
Make your half payment :
450000
Order processing is going on
////////////////////////////////////////////////////////////////////////////
Delivery order is sent to the customer
Owner name : Prashanth
Car model : ciaz
Color : red
Fuel type : petrol
Delivery date from 10 days of purchase : Thu Mar 04 17:04:42 IST 2021

Process finished with exit code 0
```

Output: (User-choice not available in Stocks)



```
"C:\Program Files\Amazon Corretto\jdk15.0.2_7\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edit
The car avaiable in Show-room
////////////////////////////////////////////////////////////////////////////
model : baleno   color : blue   engine_type : Automatic   fuel_type : petrol   unit_price : 700000
model : ciaz   color : red   engine_type : Automated_Manual_Transmission   fuel_type : petrol   unit_price : 900000
model : s-cross   color : black   engine_type : Manual   fuel_type : petrol   unit_price : 1100000
model : xl-6   color : red   engine_type : Manual   fuel_type : petrol   unit_price : 1000000
model : swift   color : white   engine_type : Automatic   fuel_type : diesel   unit_price : 500000
////////////////////////////////////////////////////////////////////////////
////////////////////////////////////////////////////////////////////////////
Enter the model :
xl-6
Enter the color :
black
Enter the engine type :
Manual
Enter the fuel type :
petrol
Make your half payment :
500000
Order processing is going on
////////////////////////////////////////////////////////////////////////////
Production order is sent to the factory
Production branch name : Avadi
Showroom address : No:47,Avadi,Chennai-54
Car model : xl-6
Color : black
Engine type : Manual
Delivery date from 20 days of purchase : Sun Mar 14 17:15:02 IST 2021
```

Hyundai-Class

```
package com.company;
import java.sql.*;
import java.util.Calendar;

public class Hyundai {
    String model;         /* baleno, ciaz, s-cross, xl-6 */
    String color;         /* red, blue, white, black */
    String engine_type;   /* Automatic, Automated Manual
Transmission(AMT), Manual */
    String fuel_type;     /* petrol, diesel */
    int unit_price;       /* price of the car from the factory
*/
    int payment;         /* payment made by the customer */

                         /* Paramterized Constructor */
    Hyundai(String data) {
        try {
            Connection con = DriverManager.getConnection
/*creating the connection*/
            ("jdbc:sqlite:C://WinterSemester-2021//CSI2008
Programming in JAVA//JAVA lab
practice//Assignment_1A_1B//cars1.db");

            Statement stm = con.createStatement(); // creating
the statem
            stm.execute("SELECT * FROM nexa WHERE model ='" +
data + "'");
            ResultSet result = stm.getResultSet();   // output
of the query is stored in result

            while (result.next()) {
                this.model = result.getString("model");
                this.color = result.getString("color");
                this.engine_type =
result.getString("engine_type");
                this.fuel_type   =
result.getString("fuel_type");
                this.unit_price  =
result.getInt("unit_price");
            }
            result.close();

            con.setAutoCommit(true);
            stm.close();
            con.close();
        }
        catch (SQLException e) {
            System.out.println("Some-thing went wrong " +
e.getMessage());
```

```
        }
    }
                    /* To display all the car details */
    public void Display() {
        System.out.println("model : " + this.model + "    " +
                "color : "      + this.color + "    " +
                "engine_type : " + this.engine_type + "    " +
                "fuel_type : "   + this.fuel_type + "    " +
                "unit_price : " + this.unit_price);
    }

                        /* Paramterized Constructor */
    Hyundai(String user_model,String user_color, String
user_engine_type,String user_fuel_type,int user_unit_price) {
        this.model = user_model;
        this.color = user_color;
        this.engine_type = user_engine_type;
        this.fuel_type   = user_fuel_type;
        this.unit_price = user_unit_price;
    }

    public void Authorize(Hyundai[] obj1) {
        int flag = 0;
        for (int i = 0; i < 5; i++) {
            if (this.unit_price == (obj1[i].unit_price/2)) {
                flag = 1;
            }
        }
        if (flag==1) {
            this.Order_Processing(obj1);
        }
        else if (flag==0){
            System.out.println("Customer, please make your
half payment");
        }
    }

    public void Order_Processing(Hyundai[] obj1) {
        System.out.println("Order processing is going on");
        int flag = 0;
            /*Checking the availability of stocks*/
        for (int i=0;i<5;i++) {
            if (
                    (this.model.equals(obj1[i].model)) &&
                    (this.color.equals(obj1[i].color)) &&

(this.engine_type.equals(obj1[i].engine_type)) &&
                    (this.fuel_type.equals(obj1[i].fuel_type))
                    /*(this.unit_price >=
(obj1[i].unit_price)/2)*/
            ) {
```

```
                flag = 1;
            }
        }

            if (flag==1) {
                this.Deliver_Order();
            }
            else {
                this.Production_Order();
            }
    }

    public void Deliver_Order() {

System.out.println("////////////////////////////////////////
////////////////////////////////");
        System.out.println("Delivery order is sent to the
customer");
        System.out.println("Owner name : Prashanth");
        System.out.println("Car model : " + this.model);
        System.out.println("Color : " + this.color);
        System.out.println("Fuel type : " + this.fuel_type);

        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.DATE, +10);
        System.out.println("Delivery date from 10 days of
purchase : " + cal.getTime());
    }

    public void Production_Order() {

System.out.println("////////////////////////////////////////
////////////////////////////////");
        System.out.println("Production order is sent to the
factory");

        System.out.println("Production branch name : Avadi");
        System.out.println("Showroom address :
No:47,Avadi,Chennai-54");
        System.out.println("Car model : " + this.model);
        System.out.println("Color : " + this.color);
        System.out.println("Engine type : " +
this.engine_type);

        Calendar cal = Calendar.getInstance();
        cal.add(Calendar.DATE, +20);
        System.out.println("Delivery date from 20 days of
purchase : " + cal.getTime());
    }
}
```

Main method

```
package com.company;

import java.sql.*;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
            try {
                                            /*  Data-base creation
*/
                Connection con = DriverManager.getConnection
/*creating the connection*/
                        ("jdbc:sqlite:C://WinterSemester-
2021//CSI2008 Programming in JAVA//JAVA lab
practice//Assignment_1A_1B//cars1.db");

                Statement stm = con.createStatement(); //
creating the statement object

                stm.execute(
                        "CREATE TABLE IF NOT EXISTS nexa" +
                                "(model text,color
text,engine_type text , fuel_type text,unit_price int)");

                stm.execute("INSERT INTO nexa VALUES
('baleno', 'blue', 'Automatic', 'petrol',700000)");
                stm.execute("INSERT INTO nexa VALUES ('ciaz',
'red', 'Automated_Manual_Transmission', 'petrol', 900000)");
                stm.execute("INSERT INTO nexa VALUES ('s-
cross', 'black', 'Manual', 'petrol', 1100000)");
                stm.execute("INSERT INTO nexa VALUES ('xl-6',
'red', 'Manual', 'petrol', 1000000)");
                stm.execute("INSERT INTO nexa VALUES ('swift',
'white', 'Automatic', 'diesel', 500000)");

                con.setAutoCommit(true);
                stm.close();
                con.close();
            }

            catch (SQLException e) {
                System.out.println("Some-thing went wrong " +
e.getMessage());
            }

        Hyundai[] obj1;
        obj1 = new Hyundai[5];

        obj1[0] = new Hyundai("baleno");
```

```
        obj1[1] = new Hyundai("ciaz");
        obj1[2] = new Hyundai("s-cross");
        obj1[3] = new Hyundai("xl-6");
        obj1[4] = new Hyundai("swift");

        /* Showing the available cars in the showroom to the
customer */
        System.out.println("The car avaiable in Show-room");

System.out.println("////////////////////////////////////////
////////////////////////////////");
        for(int i=0;i<5;i++)  {  obj1[i].Display(); }

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

        /* Getting input from the customer */
        Scanner input = new Scanner(System.in);
        String
user_model,user_color,user_engine_type,user_fuel_type;
        int user_unit_price;


System.out.println("////////////////////////////////////////
////////////////////////////////");
        System.out.println("Enter the model : ");
        user_model = input.nextLine();

        System.out.println("Enter the color : ");
        user_color = input.nextLine();

        System.out.println("Enter the engine type : ");
        user_engine_type = input.nextLine();

        System.out.println("Enter the fuel type : ");
        user_fuel_type = input.nextLine();

        System.out.println("Make your half payment : ");
        user_unit_price = input.nextInt();

        /* Creating a user object */
        Hyundai user = new
Hyundai(user_model,user_color,user_engine_type,user_fuel_type,
user_unit_price);
        user.Authorize(obj1);}
```

Github : https://github.com/PrashanthSingaravelan/WinterSemester-2021/tree/main/CSI2008%20Programming%20in%20JAVA/JAVA%20lab%20practice/Assignment_1A_1B