CSE1007
Java Programming
Lab Assignment-5

Name: Yogeswari Sahu

Registration Number: 18BCE0928

Slot: L29+L30

Faculty: KUMAR P.J

Question :

Write a program(s) in Java that illustrates the following concepts. The concepts can be applied to any practical scenario such as student administration system or software which maintains various information about students and provides various services to the users. The other scenarios may include retail business management software, banking systems, railway reservation system, online shopping applications, environmental monitoring system etc.

- Design of GUI Components using JavaFX
- Events handling
- Controls
- Database connectivity (JDBC)

Solution:

PROGRAM 1:

In a real life scenario of a finding the number of credit of a course through a form in VIT University, Read the Course name Using JavaFX, display Credit of a course in a textfield when course name is entered in another text field and user presses a button.

Concepts used:

- Design of GUI Components using JavaFX
- Events handling
- Controls

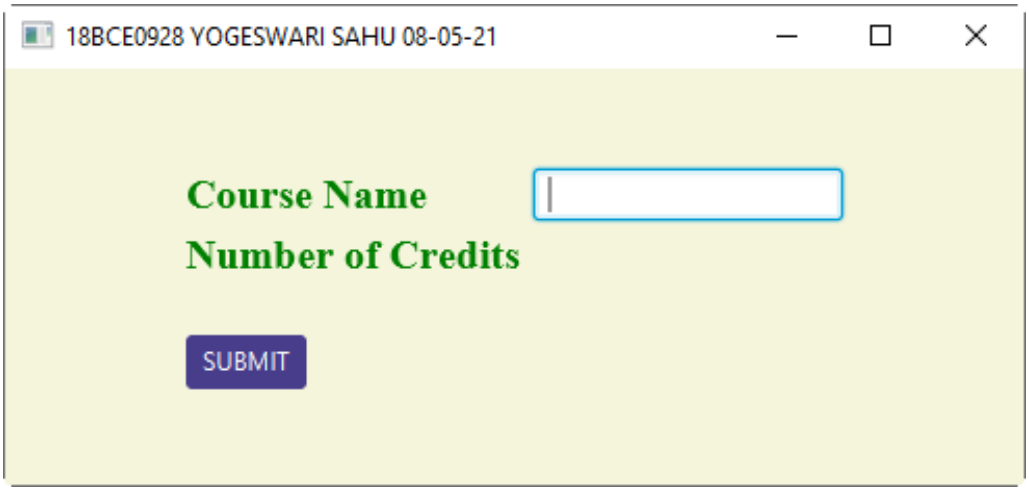
Code:

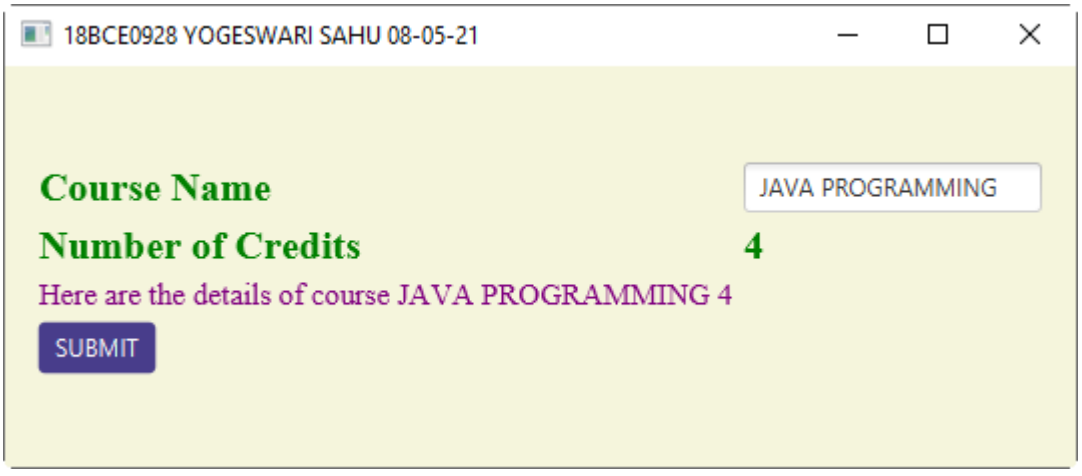
```
import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.geometry.Insets;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.control.Button;
import javafx.scene.control.PasswordField;
import javafx.scene.layout.GridPane;
import javafx.scene.text.Text;
import javafx.scene.control.TextField;
import javafx.stage.Stage;
import javafx.scene.paint.Color;
public class HelloWorld extends Application{
	@Override
	public void start(Stage stage) {
	//creating label email
	Text text1 = new Text("Course Name");
	//creating label password
	Text text2 = new Text("Number of Credits");
	Text text3 = new Text(" ");
	//Creating Text Filed for email
	TextField textField1 = new TextField();
	Text Message1 = new Text("");
	//Creating Buttons
	Button button1 = new Button("SUBMIT");
	button1.setOnAction(new EventHandler<ActionEvent>() {
	@Override
	public void handle(ActionEvent event) {
	Message1.setText("Here are the details of course "+
textField1.getText()+" "+ 4);
	text3.setText("4");
	}
	});
	//Creating a Grid Pane
	GridPane gridPane = new GridPane();
	//Setting size for the pane
	gridPane.setMinSize(400, 200);
	//Setting the padding
	gridPane.setPadding(new Insets(10, 10, 10, 10));
	//Setting the vertical and horizontal gaps between the columns
```

```
        gridPane.setVgap(5);
        gridPane.setHgap(5);
        //Setting the Grid alignment
        gridPane.setAlignment(Pos.CENTER);
        //Arranging all the nodes in the grid
        gridPane.add(text1, 0, 0);
        gridPane.add(textField1, 1, 0);
        gridPane.add(text2, 0, 1);
        gridPane.add(text3, 1, 1);
        gridPane.add(button1, 0, 3);
        gridPane.add(Message1, 0, 2);
        //Styling nodes
        button1.setStyle("-fx-background-color: darkslateblue; -fx-text-fill:
white;");
        text1.setStyle("-fx-font: normal bold 20px 'serif'; -fx-text-fill:
green; ");
        text1.setFill(Color.GREEN);
        text3.setStyle("-fx-font: normal bold 20px 'serif'; -fx-text-fill:
green; ");
        text3.setFill(Color.GREEN);
        text2.setStyle("-fx-font: normal bold 20px 'serif';-fx-text-fill:
green; ");
        Message1.setStyle("-fx-font: normal 15px 'serif';");
        Message1.setFill(Color.PURPLE);
        text2.setFill(Color.GREEN);
        gridPane.setStyle("-fx-background-color: BEIGE;");
        // Creating a scene object
        Scene scene = new Scene(gridPane);
        // Setting title to the Stage
        stage.setTitle("18BCE0928 YOGESWARI SAHU 08-05-21");
        // Adding scene to the stage
        stage.setScene(scene);
        //Displaying the contents of the stage
        stage.show();
        }
        public static void main(String args[]){
             System.out.println("Yogeswari Sahu 18BCE0928 08-05-2021");
             launch(args);
        }
}
```

Output:


18BCE0928 YOGESWARI SAHU 08-05-21
Course Name
Number of Credits
SUBMIT
HelloWorld [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (08-May-2021, 1:50:56 am)
Yogeswari Sahu 18BCE0928 08-05-2021


18BCE0928 YOGESWARI SAHU 08-05-21
Course Name
Number of Credits
SUBMIT


18BCE0928 YOGESWARI SAHU 08-05-21
Course Name
JAVA PROGRAMMING
Number of Credits
4
Here are the details of course JAVA PROGRAMMING 4
SUBMIT

PROGRAM 2:

In an online book store, there is always a need of maintaining a database for knowing the demand and quantity of books available in stock. So design a menu driven program for an admin or any user for knowing the available books, their quantity, price and their author . Design the program such that the admin can view the books to view the quantity of each book or can add any new book to the database that is in stock. Add a functionality of viewing books that are under any certain price. Establish a JDBC connection and implement the above functionalities.

Concepts used:

- Database connectivity (JDBC)
  (Operations performed- Create table, Insert records, Select Records based on values, Delete Records)

Code:

```
import java.sql.*;
import java.util.*;

public class Main {

	public static void main(String[] args) throws ClassNotFoundException {
		System.out.println("Yogeswari Sahu 18BCE0928 10-05-2021");
		Scanner sc=new Scanner(System.in);
		try{
			Class.forName("com.mysql.cj.jdbc.Driver");

			Connection
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/db","root","root
");
			System.out.println("Connection established");
			//here db is the database name, root is the user name and
root is the password
			System.out.println("Enter the choice:");
			System.out.println("1- get column properties");
			System.out.println("2-insert values into table");
			System.out.println("3-deleting values");
			System.out.println("4-showing all records");
			System.out.println("5-showing values below the particular
price");
```

```
                int choice = sc.nextInt();
switch (choice) {

                case 1:
                      try {

                         Statement st = con.createStatement();
                         ResultSet rs = st.executeQuery("select * from
books");
                            ResultSetMetaData rsmd = rs.getMetaData();
                            int columnCount = rsmd.getColumnCount();
                            System.out.println("Number of Columns:
"+columnCount);
                            for(int i=1;i<=columnCount;i++){
                                  System.out.println("Column Name :
"+rsmd.getColumnName(i)+", Column Type: "+rsmd.getColumnType(i));
                            }
                      } catch (SQLException e) {
                            e.printStackTrace();
                      }
                      break;

                case 2:
                      try {
                            sc.nextLine();
                            //input Book ID
                            System.out.println("Enter Book ID: ");
                            String id=sc.nextLine();
                            //input Author name
                            System.out.println("Enter Title of Book : ");
                            String title=sc.nextLine();
                            //input Title of Book
                            System.out.println("Enter Author Name : ");
                            String author=sc.nextLine();
                            //input Price
                            System.out.println("Enter Price: ");
                            String price=sc.nextLine();
                            //input Quantity
                            System.out.println("Enter Quantity: ");
                            String qty=sc.nextLine();

                            //creating object of PreparedStatement class and
passing parameter (?)
                            PreparedStatement smt=con.prepareStatement("insert
into books values(?,?,?,?,?)");

                            // set the values
                            smt.setInt(1, Integer.parseInt(id));
                            smt.setString(2, title);
```

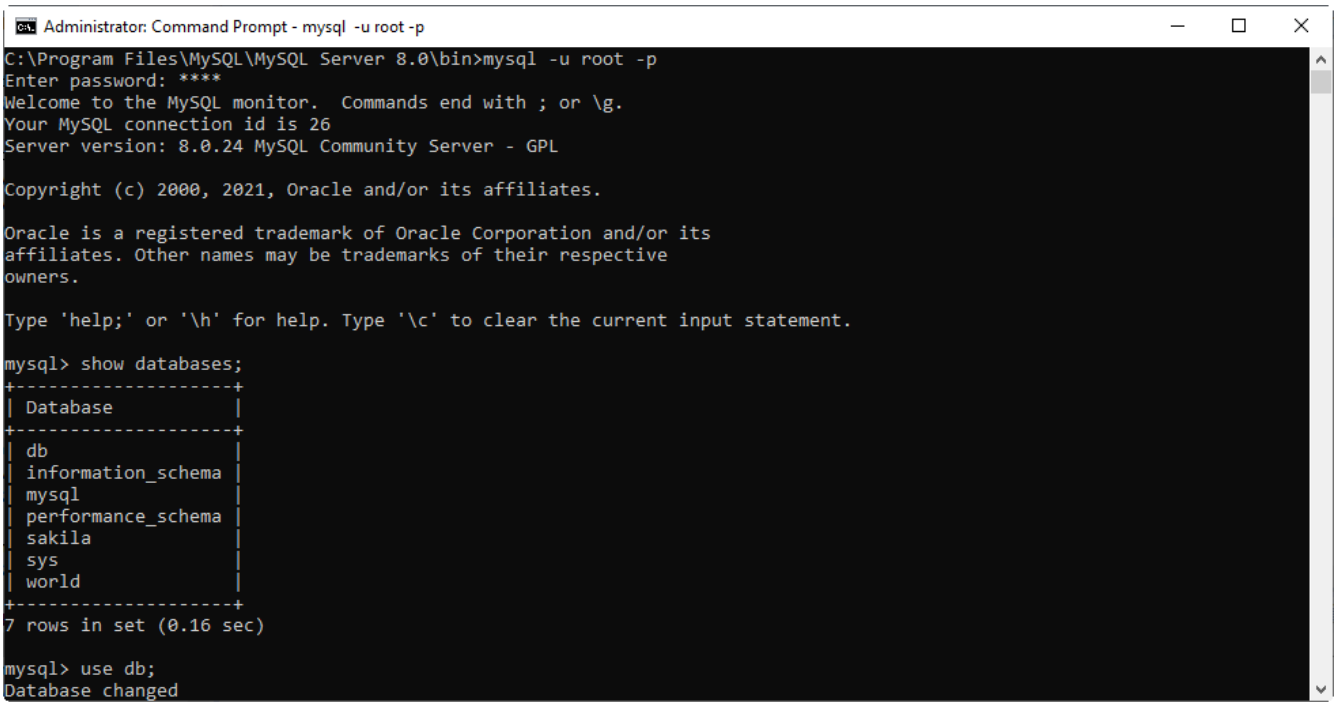
```
                        smt.setString(3, author);
                        smt.setDouble(4, Double.parseDouble(price));
                        smt.setInt(5, Integer.parseInt(qty));

                        //to execute update
                        smt.executeUpdate();
                        System.out.println("Record Inserted....");
                    }catch(Exception e) {
                        e.printStackTrace();
                    }
                    break;

                case 3:
                        try{
                        Statement stmt=con.createStatement();
                        //input a particular Book id of which we want to
delete record
                        System.out.print("Enter Book ID:");
                        String bid=sc.next();

                        //query to take data of a particular record from
table books
                        String q="Select * from books where id='"+bid+"'";

                        //to execute query
                        ResultSet rs=stmt.executeQuery(q);

                        if(rs.next())
                        {
                            //to show the data
                            System.out.println("Book id: "+rs.getInt(1));
                            System.out.println("Book title
:"+rs.getString(2));
                            System.out.println("Book author
:"+rs.getString(3));
                            System.out.println("Book
price:"+rs.getDouble(4));
                            System.out.println("Sure To Delete Above
Record Yes/No?");
                            String ch=sc.next();
                            if(ch.equalsIgnoreCase("yes"))
                            {
                                //query to delete data of a particular
record from table books
                                q="delete from books where
id='"+bid+"'";
                                //to execute query
                                stmt.executeUpdate(q);
```

```
                                        System.out.println("Record
Deleted...");
                                  }
                            }
                            else
                            {
                                  System.out.println("Record Not Found...");
                            }
                      }
                      catch(Exception e){
                            System.out.println(e);
                      }
                      break;

                case 4:
                      try {
                            Statement stmt = con.createStatement();
                            String strSelect = "select * from books";
                            ResultSet rs = stmt.executeQuery(strSelect);
                            System.out.println("The records selected are:");
                         int rowCount = 0;
                         System.out.println("Book id\tBook title\t\tBook
author\tBook price\tBook quantity");
                        while(rs.next()) {   // Repeatedly process each row

      System.out.println(rs.getInt(1)+"\t"+rs.getString(2)+"\t"+rs.getString(3
)+"\t"+rs.getDouble(4)+"\t\t"+rs.getInt(5));
                             ++rowCount;
                         }
                         System.out.println("Total number of records = " +
rowCount);
                      }catch(Exception e) {
                            e.printStackTrace();
                      }
                      break;
                case 5:
                      try {
                            System.out.print("Enter Price: ");
                            String price=sc.next();
                            PreparedStatement smt=con.prepareStatement("select
* from books where price >= ?");
                            smt.setDouble(1, Double.parseDouble(price));
                            ResultSet rs = smt.executeQuery();
                            System.out.println("The records selected are:");
                         int rowCount = 0;
                         while(rs.next()) {   // Repeatedly process each row
                                        System.out.println("Book id:
"+rs.getInt(1)+",\tBook title: "+rs.getString(2)+",\tBook Author:
```

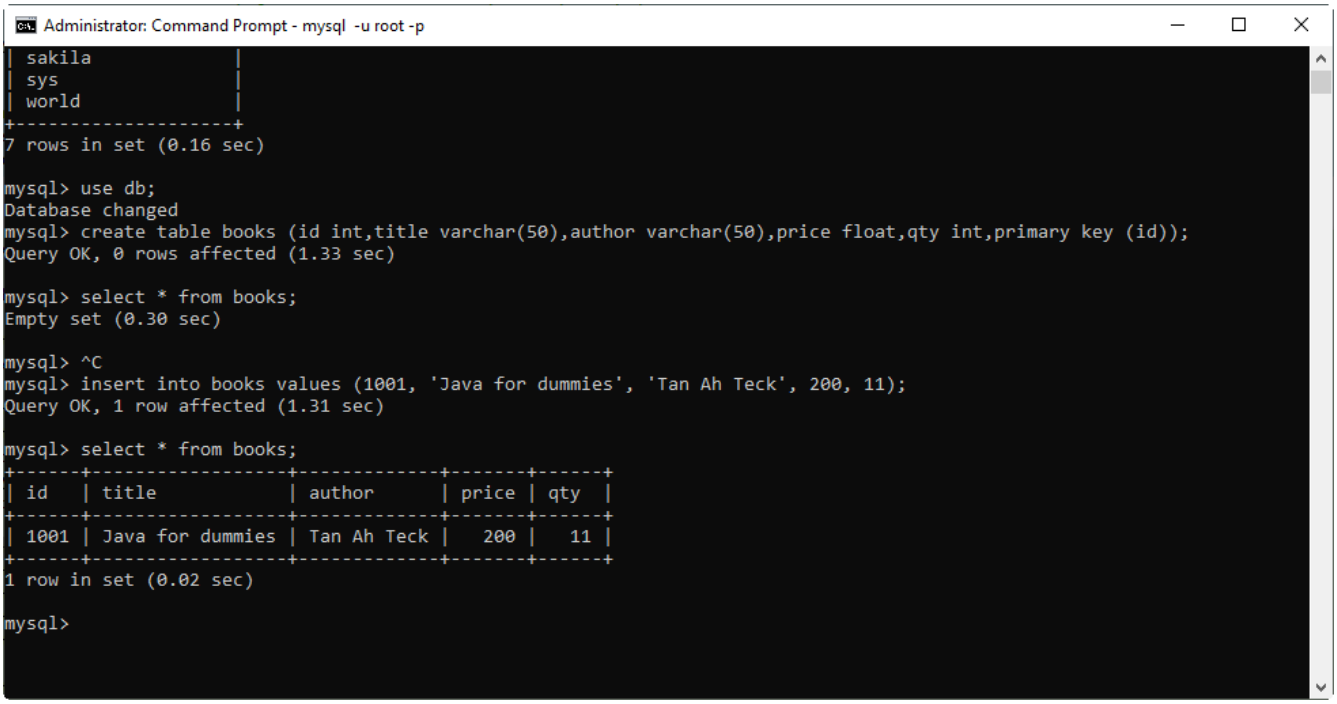
```
"+rs.getString(3)+",\tBook price: "+rs.getDouble(4)+",\tBook quantity:
"+rs.getInt(5));
                              ++rowCount;
                           }
                           System.out.println("Total number of records where
price >="+price+": "+ rowCount);
                        }catch(Exception e) {
                              e.printStackTrace();
                        }
                        break;
                  default:
                        System.out.println("Please enter a number from 1-5");

}
                  con.close();
                  }catch(Exception e){ System.out.println(e);}
                  sc.close();
      }
}
```

Output:

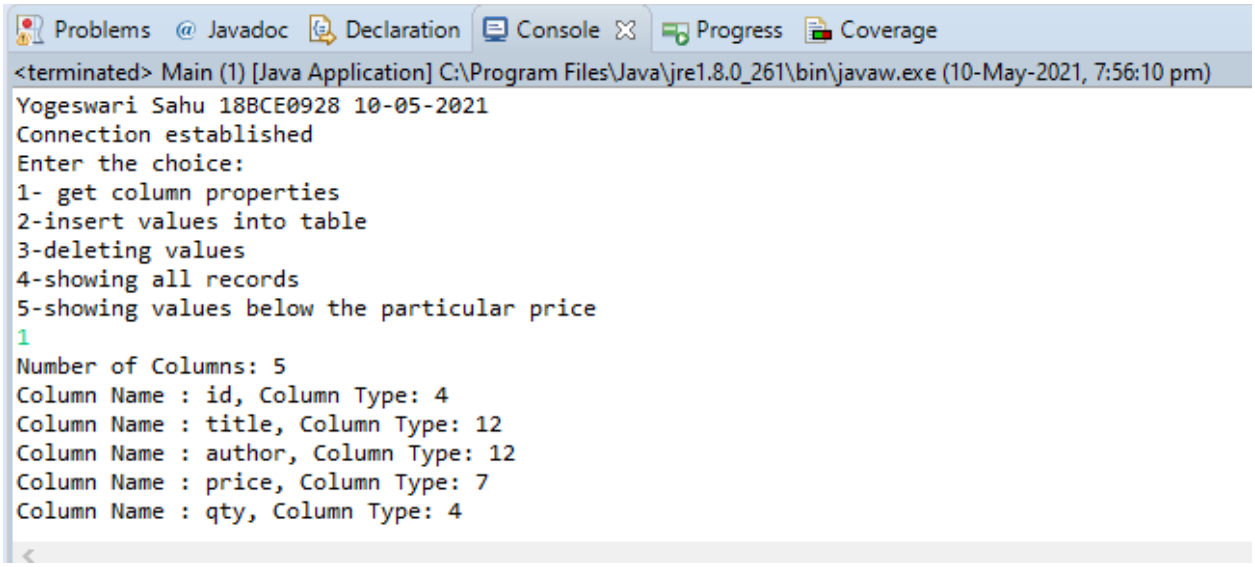
Creating a table in the database:



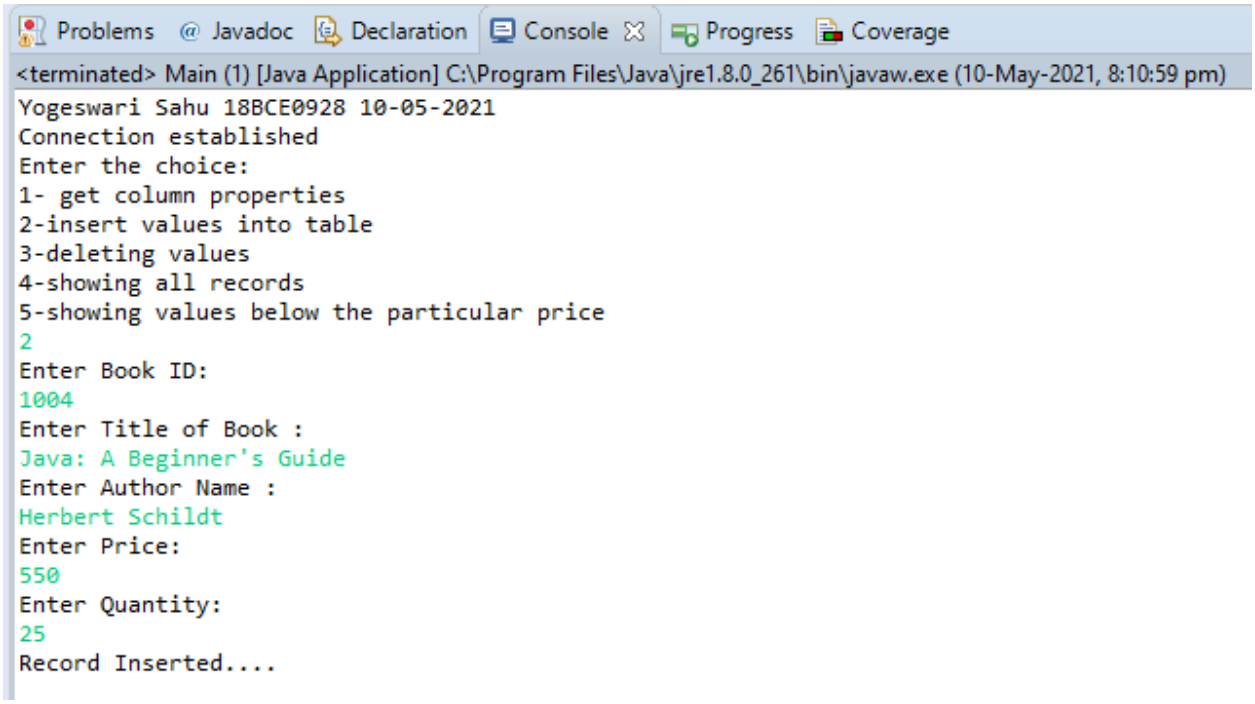
View the records:

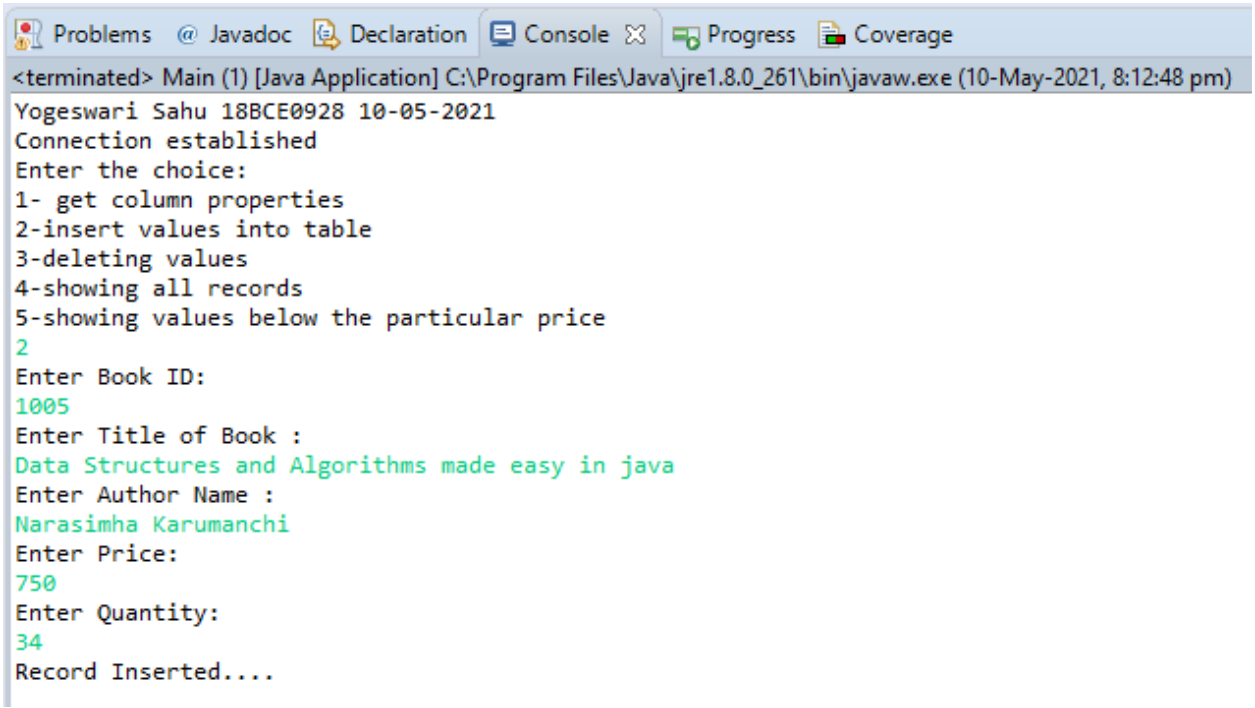


To get all the column properties:



Insert records to the table:





Viewing all the records after insertion of records:

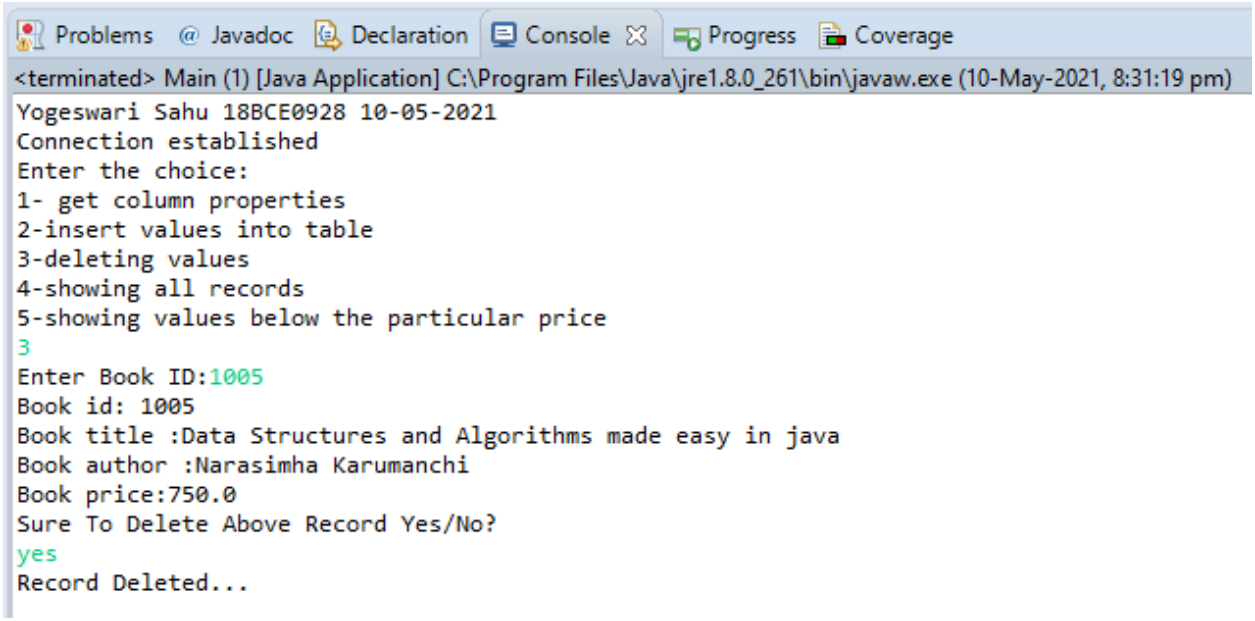
```
Problems  Javadoc  Declaration  Console  Progress  Coverage
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (10-May-2021, 8:17:15 pm)
1- get column properties
2-insert values into table
3-deleting values
4-showing all records
5-showing values below the particular price
4
The records selected are:
Book id Book title                  Book author     Book price      Book quantity
1001    Java for dummies            Tan Ah Teck     200.0           11
1002    More Java for dummies       Tan Ah Teck     250.0           5
1003    A Teaspoon of Java          Kevin Jones     430.0           25
1004    Java: A Beginner's Guide            Herbert Schildt 550.0           25
1005    Data Structures and Algorithms made easy in java        Narasimha Karumanchi    750.0           34
Total number of records = 5
```

```
Administrator: Command Prompt - mysql  -u root -p

mysql> select * from books;
+------+-----------------------------------------------+---------------------+-------+------+
| id   | title                                         | author              | price | qty  |
+------+-----------------------------------------------+---------------------+-------+------+
| 1001 | Java for dummies                              | Tan Ah Teck         |   200 |   11 |
| 1002 | More Java for dummies                         | Tan Ah Teck         |   250 |    5 |
| 1003 | A Teaspoon of Java                            | Kevin Jones         |   430 |   25 |
| 1004 | Java: A Beginner's Guide                      | Herbert Schildt     |   550 |   25 |
| 1005 | Data Structures and Algorithms made easy in java | Narasimha Karumanchi |   750 |   34 |
+------+-----------------------------------------------+---------------------+-------+------+
5 rows in set (0.00 sec)

mysql>
```

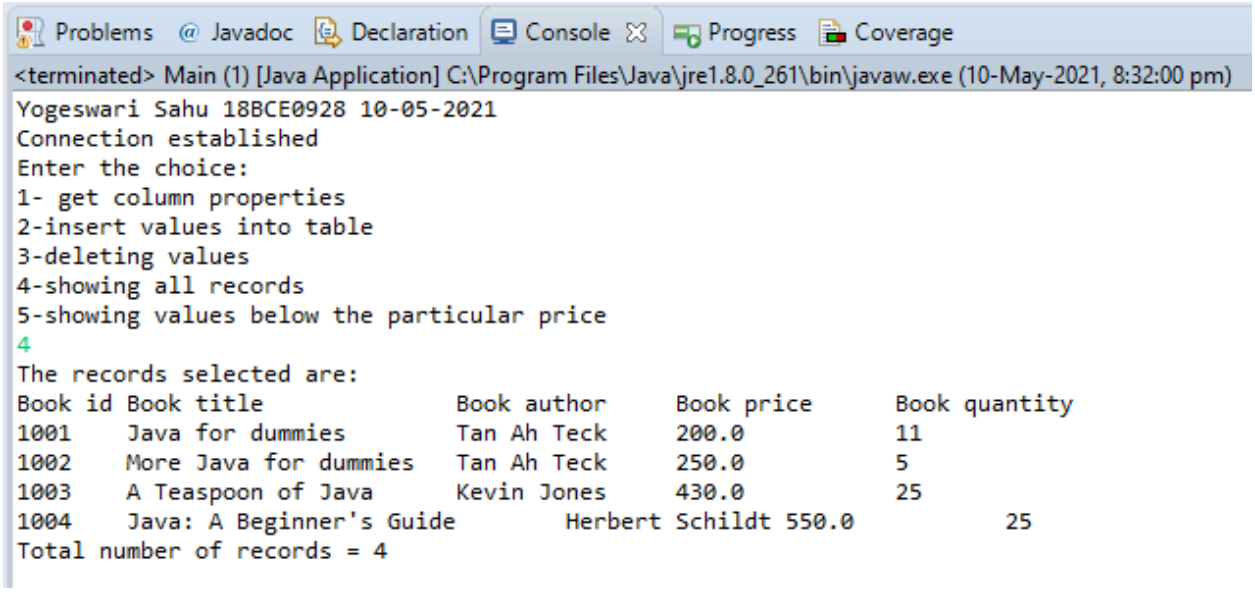
Showing records above a particular price given by the user:

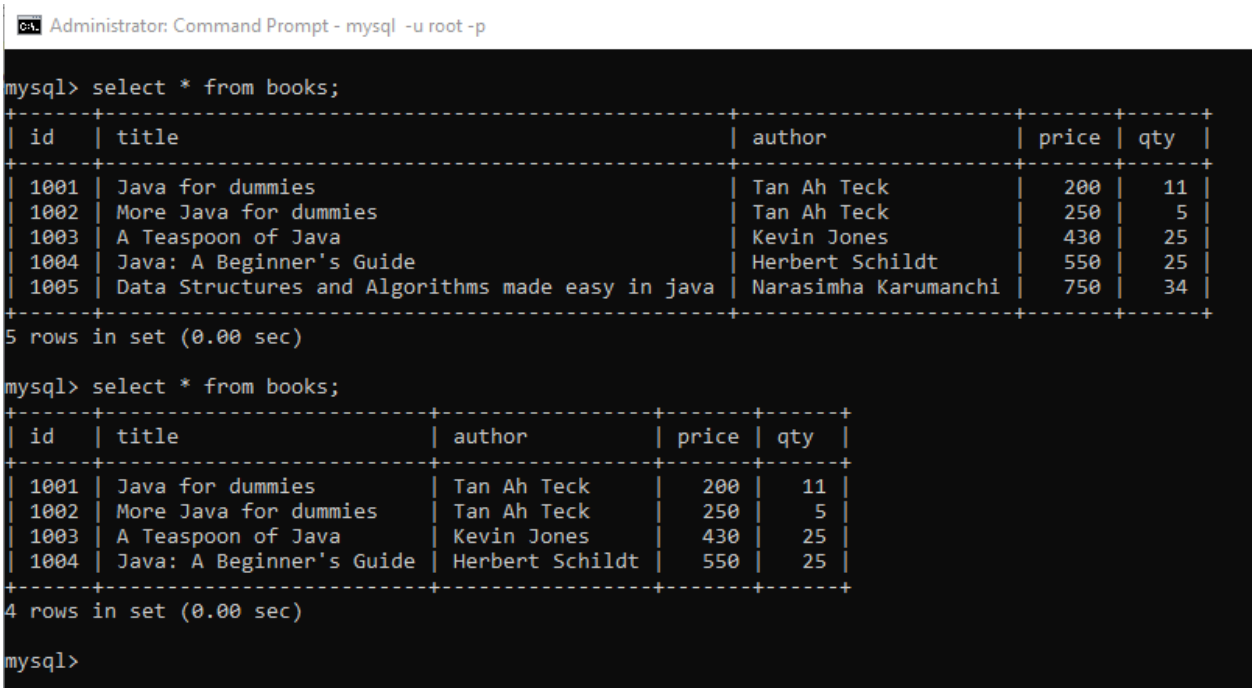
```
Problems  Javadoc  Declaration  Console  Progress  Coverage
<terminated> Main (1) [Java Application] C:\Program Files\Java\jre1.8.0_261\bin\javaw.exe (10-May-2021, 8:28:26 pm)
Yogeswari Sahu 18BCE0928 10-05-2021
Connection established
Enter the choice:
1- get column properties
2-insert values into table
3-deleting values
4-showing all records
5-showing values below the particular price
5
Enter Price: 400
The records selected are:
Book id: 1003,  Book title: A Teaspoon of Java, Book Author: Kevin Jones,       Book price: 430.0,      Book quantity: 25
Book id: 1004,  Book title: Java: A Beginner's Guide,   Book Author: Herbert Schildt,   Book price: 550.0,      Book quantity: 25
Book id: 1005,  Book title: Data Structures and Algorithms made easy in java,   Book Author: Narasimha Karumanchi,      Book price: 750.0,      Book
quantity: 34
Total number of records where price >=400: 3
```

Deleting a record specified by the user:



Viewing the records after deletion of the record:




Administrator: Command Prompt - mysql -u root -p
mysql> select * from books;
+------+---+----------------------+-------+------+
| id | title | author | price | qty |
+------+---+----------------------+-------+------+
1001	Java for dummies	Tan Ah Teck	200	11
1002	More Java for dummies	Tan Ah Teck	250	5
1003	A Teaspoon of Java	Kevin Jones	430	25
1004	Java: A Beginner's Guide	Herbert Schildt	550	25
1005	Data Structures and Algorithms made easy in java	Narasimha Karumanchi	750	34
+------+---+----------------------+-------+------+
5 rows in set (0.00 sec)
mysql> select * from books;
+------+--------------------------+-----------------+-------+------+
| id | title | author | price | qty |
+------+--------------------------+-----------------+-------+------+
1001	Java for dummies	Tan Ah Teck	200	11
1002	More Java for dummies	Tan Ah Teck	250	5
1003	A Teaspoon of Java	Kevin Jones	430	25
1004	Java: A Beginner's Guide	Herbert Schildt	550	25
+------+--------------------------+-----------------+-------+------+
4 rows in set (0.00 sec)
mysql>