CIS355A Lab Report

**Your name: Don Killen**

**Date: October 18, 2015**

**Lab week \_\_\_Week 7 Final Project: 1. program code, 2. User Manual attached below.**

**Objective/Purpose of the program**

Program demonstrates a flooring application called Floored.java, using MySQL database to print up a list of entered customers from the application.

**Analysis/Design**

[Describe the approach / structure of program. What classes/functions were used? ] Five tabs were used to narrow the number of GUI’s per tab and entered into the database. Functions added for connecting to the database, creating the GUI’s, creating the table for the database, entering a query into the database to enter customer data and finally close the connection and database. Additionally, a driver was used that includes main().

**Testing/Results**

[Does your program satisfy all requirements of the lab? yes

If yes, how did you test it?  (Indicate test cases used, expected values, and **show results with screen shots) See screenshots below. Validation added for input of customer information and area calculation for length and width, double values. (See User Manual, page 17, for output of test data)**

If any requirements are NOT met, document the known issues. What did you do to try to solve them?

Make sure you demonstrate in your testing the parts that are working correctly.]

**Conclusions / Lessons Learned**

[What difficult problems did you encounter and how did you handle them? Review java for validation of data for inputs. Table error for table space missing: drop table used to recreate table.

What new concepts did you learn/reinforce with this lab? I Learned to use MySQL and JDBC connection configurations. Also learned to use Statements and Prepared statements to handle database queries.

Is there anything you would have done differently?] Keep program more modular and less complex, so less prone to errors.

Screen shots of source code and main (driver):

1. Floored.java:

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Program Name: Floored.java

Programmer's Name: Don Killen

Program Description: Develop an application for a flooring company with two types of flooring: Carpet and Wood

The user will enter their name, address, type of flooring, area, review the order, then submit it to the MySQL

database called flooringdb and table called flooring. The area is calculated and then the cost. Then the order is sent to the

database or user exits, or clears the entry and starts over again. If they submit the order a list of orders are printed out.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

import java.awt.GridBagLayout;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.\*;

import javax.swing.\*;

import java.sql.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.sql.Statement;

public class Floored implements ActionListener{

private static final String DATABASE\_NAME = "flooringdb";

static Connection connection = null;

static Statement statement = null;

static ResultSet resultSet = null;

private JTextField addressField = new JTextField();

private JTextField nameField = new JTextField();

private JTextField lengthField = new JTextField();

private JTextField widthField = new JTextField();

private JButton areaButton = new JButton();

private JButton clearButton = new JButton();

private JRadioButton wood= new JRadioButton();

private JRadioButton carpet = new JRadioButton();

private JButton costButton = new JButton();

private JButton displayButton = new JButton();

private JButton exitButton = new JButton();

private JButton submitButton = new JButton();

private static String name;

private static String strAddress;

private static double length;

private static double width;

private static double area = 0.00 ;

private static double cost = 0.00 ;

private static double floorCost;

private static String floorType = "";

public void createGUI(){

// create Frame

JFrame frame = new JFrame();

// create 4 Panels and 1 Layout

JPanel p = new JPanel(new GridBagLayout());

frame.getContentPane().add(p, BorderLayout.NORTH);

JPanel p2 = new JPanel();

JPanel p3 = new JPanel();

JPanel p4 = new JPanel();

JPanel p5 = new JPanel ();

//Set up Dimensions

frame.getContentPane().add(p, BorderLayout.NORTH);

frame.getContentPane().setBackground(Color.BLUE);

frame.setVisible(true);frame.setSize(500,300);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setTitle("\*Welcome to 'The Floored Company'\*");

//Set Constraints

GridBagConstraints c = new GridBagConstraints();

//This will position the various components.

//JTabbedPane (5 tabs)

JTabbedPane j = new JTabbedPane();

j.add("#1 Customer Info",p);

j.add("#2 Flooring Style", p2);

j.add("#3 Dimensions", p3);

j.add("#4 Calculations", p4);

j.add("#5 Display Summary",p5);

frame.add(j);

//First Panel

//Name Label

JLabel nameLabel = new JLabel("Customer Name:");

c.gridx=0;

c.gridy=0;

p.add(nameLabel,c);

c.insets = new Insets(10,10,10,10);

//JTextField for Name

nameField = new JTextField(20);

nameField.setEditable(true);

nameField.addActionListener(this);

c.gridx=1;

c.gridy=0;

p.add(nameField,c);

c.insets = new Insets(10,10,10,10);

//Address Label

JLabel addressLabel = new JLabel("Customer Address:");

c.gridx=0;

c.gridy=2;

p.add(addressLabel,c);

c.insets = new Insets(10,10,10,10);

//JTextField for Address

addressField = new JTextField(45);

addressField.setEditable(true);

addressField.addActionListener(this);

c.gridx=1;

c.gridy=2;

p.add(addressField,c);

c.insets = new Insets(10,10,10,10);

// Second Panel

//Floor Label

JLabel floorType = new JLabel ("What Type of Floor?");

c.gridx=0;

c.gridy=0;

p2.add(floorType,c);

c.insets = new Insets(10,10,10,10);

//Radio Buttons

wood= new JRadioButton("Wood");

c.gridx=0;

c.gridy=3;

p2.add(wood,c);

wood.addActionListener(this);

c.insets = new Insets(10,10,10,10);

JLabel woodLabel = new JLabel ("$20");

c.gridx=0;

c.gridy=4;

p2.add(woodLabel,c);

c.insets = new Insets(10,10,10,10);

carpet= new JRadioButton("Carpet");

c.gridx=1;

c.gridy=3;

p2.add(carpet,c);

carpet.addActionListener(this);

c.insets = new Insets(10,10,10,10);

JLabel carpetLabel = new JLabel ("$10");

c.gridx=1;

c.gridy=4;

p2.add(carpetLabel,c);

c.insets = new Insets(10,10,10,10);

// Third Panel

//Floor Width Label

JLabel widthLabel= new JLabel ("Please enter the WIDTH of your chosen floor:");

c.gridx=0;

c.gridy=0;

p3.add(widthLabel,c);

c.insets = new Insets(10,10,10,10);

//Width TextField

//JTextField for Name

widthField = new JTextField(20);

widthField.setEditable(true);

widthField.addActionListener(this);

c.gridx=1;

c.gridy=0;

p3.add(widthField,c);

c.insets = new Insets(10,10,10,10);

//Length Label

JLabel lengthLabel= new JLabel ("Please enter the LENGTH of your choice of floors:");

c.gridx=1;

c.gridy=0;

p3.add(lengthLabel,c);

c.insets = new Insets(10,10,10,10);

//JTextField Length

lengthField = new JTextField(20);

lengthField.setEditable(true);

lengthField.addActionListener(this);

c.gridx=1;

c.gridy=2;

p3.add(lengthField,c);

c.insets = new Insets(10,10,10,10);

// Fourth Panel

JLabel calcAreaLabel = new JLabel(" Calculate The Area: click here");

c.gridx=0;

c.gridy=0;

p4.add(calcAreaLabel,c);

c.insets = new Insets(10,10,10,10);

areaButton = new JButton("Calc Area");

c.gridx=0;

c.gridy=1;

p4.add(areaButton,c);

c.insets = new Insets(10,10,10,10);

areaButton.addActionListener(this);

JLabel calcCostLabel = new JLabel(" Calculate The Cost: click here");

c.gridx=0;

c.gridy=2;

p4.add(calcCostLabel,c);

c.insets = new Insets(10,10,10,10);

costButton = new JButton("Calc Cost");

c.gridx=0;

c.gridy=3;

p4.add(costButton,c);

c.insets = new Insets(10,10,10,10);

costButton.addActionListener(this);

//Fifth Panel

JLabel displayLabel = new JLabel(" Display Order: click here");

c.gridx=0;

c.gridy=0;

p5.add(displayLabel,c);

c.insets = new Insets(10,10,10,10);

displayButton = new JButton("Review Your Order");

c.gridx=0;

c.gridy=1;

p5.add(displayButton,c);

c.insets = new Insets(10,10,10,10);

displayButton.addActionListener(this);

JLabel submitLabel = new JLabel(" Submit Your Order: click here");

c.gridx=0;

c.gridy=2;

p5.add(submitLabel,c);

c.insets = new Insets(10,10,10,10);

submitButton = new JButton("SUBMIT");

c.gridx=0;

c.gridy=3;

p5.add(submitButton,c);

c.insets = new Insets(10,10,10,10);

submitButton.addActionListener(this);

clearButton = new JButton("Clear");

c.gridx=2;

c.gridy=3;

p5.add(clearButton,c);

c.insets = new Insets(10,10,10,10);

clearButton.addActionListener(this);

exitButton = new JButton("Exit");

c.gridx=1;

c.gridy=3;

p5.add(exitButton,c);

c.insets = new Insets(10,10,10,10);

exitButton.addActionListener(this);

}

public static void connectTodb(){

try //check for valid connection

{ // external Jar file used to connect to db

System.out.println("Connecting to Driver....");

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

System.out.println("Connected to Driver; Success!");

//Connect to database from driver

System.out.println("Connecting to: " + DATABASE\_NAME + "..........");

connection = DriverManager.getConnection( "jdbc:mysql://localhost:3306", "root", "bmu33@R45sD");

System.out.println("Connected to " + DATABASE\_NAME + " Database; \nSuccess!");

}

catch (ClassNotFoundException error) //invalid connection

{

System.out.println("Connection Failed: " + error.getMessage()); //tell the user of an error

}

catch (SQLException error)

{

System.out.println("ERROR: " + error.getMessage());

}

}

// Action event handler

@Override

public void actionPerformed(ActionEvent e) {

if(e.getSource() == wood){

cost = 20;

floorType = "Wood";

}

if(e.getSource() == carpet)

{

cost = 10;

floorType = "Carpet";

}

if(e.getSource() == areaButton)

{

String l = lengthField.getText();

String w = widthField.getText();

calculateArea(l,w);

}

if (e.getSource() == costButton)

{

floorCost = cost \* area;

}

if (e.getSource()==displayButton)

{ name = nameField.getText();

// check for valid input of name

if(nameField.getText().equals(""))

{

JOptionPane.showMessageDialog(null, "must enter a name" );

return;

}

strAddress = addressField.getText();

// check for valid input of address

{

if(addressField.getText().equals("")){

JOptionPane.showMessageDialog(null, "must enter an address" );

return;

}

}

String str = "Name: " + name + "\n" + "Address: " + strAddress + "\n" + "Floor: " + floorType + "\n" + "Area: " + area + "sq.ft"+"\n" + "Total Cost: " +"$"+ floorCost;

JOptionPane.showMessageDialog(null,str);

}

if(e.getSource()==submitButton)

{

saveTodb();

}

if (e.getSource()==clearButton)

{

nameField.setText("");

addressField.setText("");

floorType = "";

area = 0.0;

cost = 0.0;

floorCost = 0.0;

lengthField.setText("");

widthField.setText("");

carpet.setSelected(false);

wood.setSelected(false);

}

if (e.getSource() == exitButton)

{

System.exit(0);

}

}

public static void saveTodb()

/\* Besides saving data to database it includes four other methods that link here:

\* 1. connectToDD();

\* 2. createTables(); only one table "flooring"

\* 3. sqlQuery(); perform queries

\* 4. closeConnection(); close the database

\*/

{

JOptionPane.showMessageDialog(null, "Saving to database....");

connectTodb();

createTables();

sqlQuery();

closeConnection();

}

public static void createTables()

//recreate database table "flooring"

{

try

{

statement = connection.createStatement();

//statement.execute("DROP TABLE flooringdb.flooring"); //not needed except to clear table manually

statement.execute("CREATE TABLE flooringdb.flooring ( name VARCHAR(30) NOT NULL, address VARCHAR(45) NOT NULL PRIMARY KEY , "

+ " floorType VARCHAR(30) NOT NULL, area DOUBLE(20,2) NOT NULL, floorCost DOUBLE(20,2) NOT NULL )");

System.out.println("Table: flooring has been created");

}

catch (SQLException error)

{

System.out.println("ERROR " + error.getMessage());

}

} //End createTable

public static void sqlQuery()

// create query to insert customer info into database table

{

try

{

// create Insert Statements for customer info, area and choice of flooring cost

PreparedStatement statement = connection.prepareStatement("INSERT INTO flooringdb.flooring (name, address, floorType, area, floorCost) VALUES (?,?,?,?,?)");

statement.setString(1, name);

statement.setString(2, strAddress);

statement.setString(3, floorType);

statement.setDouble(4, area);

statement.setDouble(5, floorCost);

statement.executeUpdate();

System.out.println("Data inserted into database");

ResultSet results = statement.executeQuery("Select \* from flooringdb.flooring");

while(results.next()) // loop to add to and print out table flooring list

{

System.out.println("Name: " + results.getString(1) + " Address: " + results.getString(2) + " " +

" Floor Type: " + results.getString(3) + " Area: " + results.getString(4) + " Flooring Cost: $" +

results.getString(5));

}

}

catch (SQLException error)

{

System.out.println("ERROR: " + error.getMessage());

}

}

public static void closeConnection()

// make sure connection is closed properly

{

if (connection !=null)

{

try

{

connection.close();

System.out.println("Connection CLOSED");

}

catch (SQLException ignore)

{

System.out.println("ERROR: could not close connection");

}

}

if (statement !=null)

{

try

{

statement.close();

System.out.println("Statment closed");

}

catch (SQLException ignore)

{

System.out.println("ERROR: could not close statement");

}

}

}

public void calculateArea(String l, String w) {

//calculate cost based on dimensions of area and parse double fields

l = lengthField.getText();

w = widthField.getText();

// catch length not entered as a double

try

{

Double.parseDouble(l);

}

catch(NumberFormatException e)

{ //not a double

JOptionPane.showMessageDialog(null, "must input a number for length");

return;

}

try

{

Double.parseDouble(w);

}

catch(NumberFormatException e)

{ //not a double

JOptionPane.showMessageDialog(null, "must input a number for width");

return;

}

length =Double.parseDouble(l);

width = Double.parseDouble(w);

area = length \* width;

}

}

1. testFloored.java: (driver, main) source code:

**public** **class** testFloored {

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

Floored p = **new** Floored();

p.*connectTodb*();

p.createGUI();

}

}

Don Killen

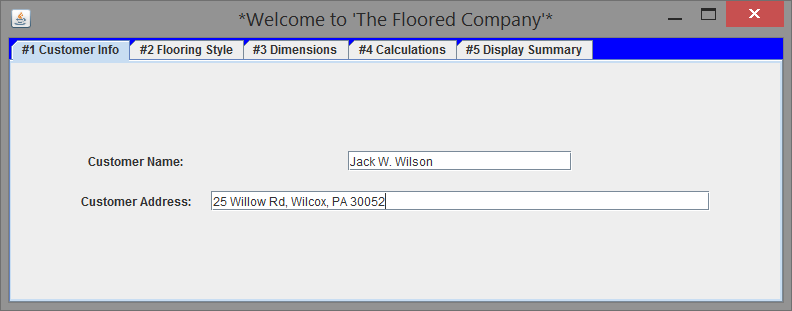
CIS 355A

October 18, 2015

Final Project: User Manual, the Floored Company (a mock flooring company)

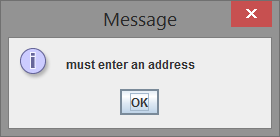
This manual will demonstrate the various functions of the flooring application for the Floored Company using screenshots below.

The first tab of the five tabs is for the customer information and includes enough room for those long addresses:



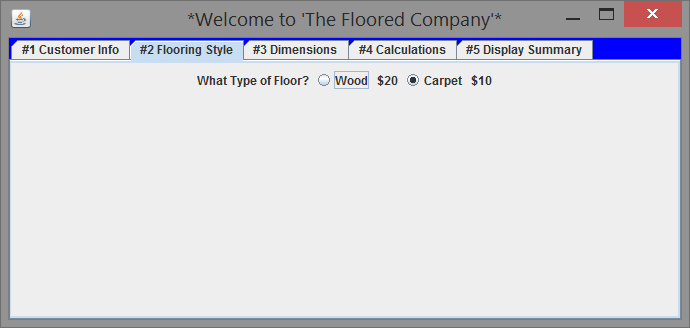
Validation for customer information: if no name or address entered corresponding messages list the error to the user below:



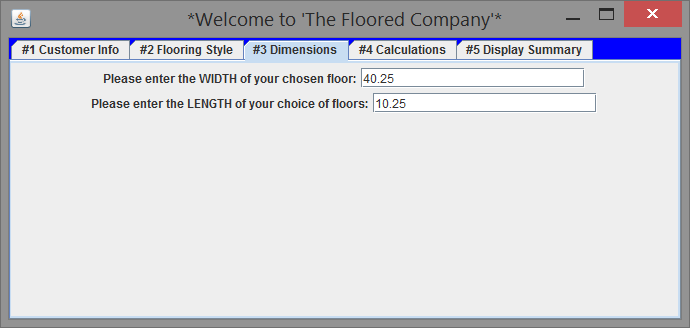


The user must go back to the CUSTOMER INFO tab and enter the missing information.

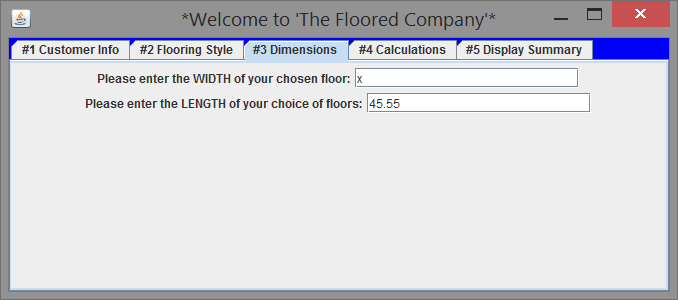
The next tab, tab #2, is the customer’s choice of flooring, either wood or carpet with the price listed for each, $20 for wood and $10 for carpet:



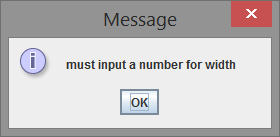
Tab #3 is for listing the dimensions of the area to be calculated using double values:



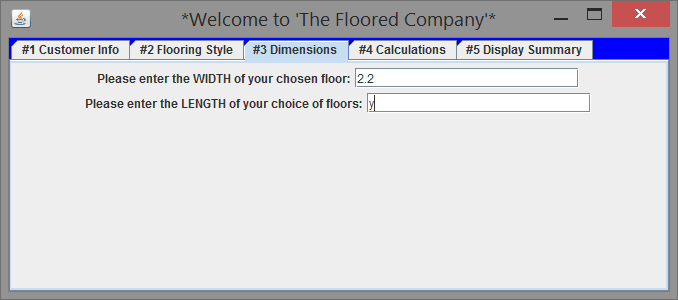
Data validation: If a number isn’t entered for one or more fields an error message is output below during CALCULATIONS in the next tab:

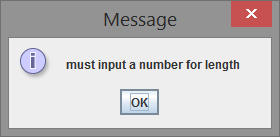


Wrong data entered for WIDTH above and corresponding message below



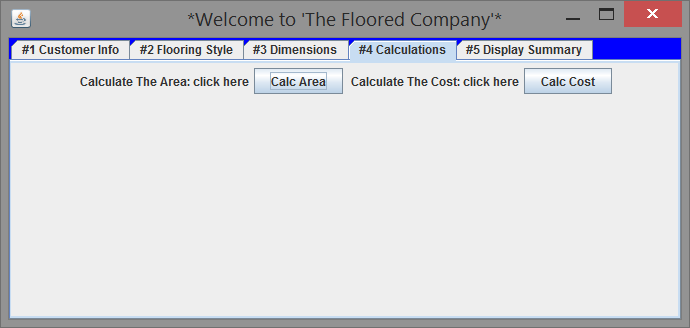
Error message for validation of data entered for width if not a double value above message prints out for corresponding length below.



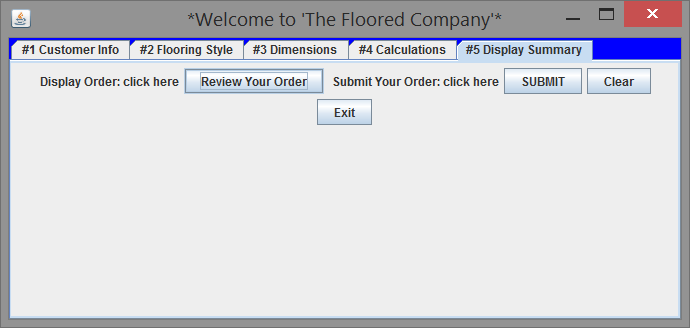


Error message: for validation of input for double for length above.

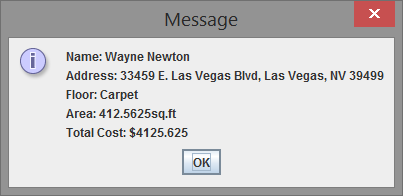
Tab #4 calculates the cost by using the values entered to total the cost by clicking the Calc area button and then Calc cost in consecutive left to right order as normally done: In order to see the calculations summarized click the next tab to display the summary.



Tab #5: When the user clicks the Review Your Order Button a pop-up message displays the initial order (see below tab #5 below)

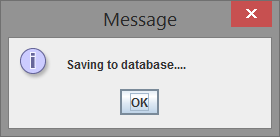


The message screen appears below and the user presses OK.



If the user is satisfied with the order they hit the SUBMIT button on the right or hit CLEAR to start over or hit the Exit button to log out of the system completely.

If the user continues by hitting the SUBMIT button the following screen appears:



Saving the customer order to the database and the following formatted output is listed in the console output:

Connecting to Driver....

Connected to Driver; Success!

Connecting to: flooringdb..........

Connected to flooringdb Database;

Success!

Connecting to Driver....

Connected to Driver; Success!

Connecting to: flooringdb..........

Connected to flooringdb Database;

Success!

ERROR Table 'flooring' already exists

Data inserted into database

Name: Bob Marley Address: 2520 Island Way, Sarasota, FL 30053 Floor Type: Wood Area: 463.05 Flooring Cost: $9261.06

Name: Gregory Peck Address: 2959 Exclusive Dr, Los Angeles, CA 20045 Floor Type: Wood Area: 813.85 Flooring Cost: $16277.10

Name: Wayne Newton Address: 33459 E. Las Vegas Blvd, Las Vegas, NV 39499 Floor Type: Carpet Area: 412.56 Flooring Cost: $4125.62

Name: Mathew McConahay Address: 4590 Kangaroo Way, Maui, HI 40025 Floor Type: Wood Area: 893.20 Flooring Cost: $17864.00

Connection CLOSED

Statment closed

The customer’s data is listed by title; and area and cost are properly formatted within two decimal places, $ 0.00 (placeholder) cost and added to the list of customers which were already written to the database previously to show the list works properly. Additional information lists when the connection to the database is achieved and the database name, “flooringdb”. The table “flooring” was created again showing an error, but adds the data despite the warning because a “drop table” command is used to clear the table if necessary.