

## **Criteria C: Development**

Word count: 770

### **Introduction:**

The product is a Java program run on NetBeans and MySQL. It also uses Java's Swing tools to make the Graphical User Interface. The program accepts input from my client about what kind of outfit he wants to wear based on criteria they set, and they also have the option to save the outputted outfit. Additionally, my client is able to add and delete clothing items from their virtual closet. It searches for articles of clothing that fit the users input and produces 2 outfit options that fit the specified criteria.

### **List of techniques used:**

- Random number generation
- For loop
- While loop
- Array list
- Objects
- Inheritance
- Try statements
- Catch statements
- Simple selection (if statements)
- Option pane generation for communicating with the user
- GUI popup menus
- Nested loops
- MySQL database connections
- Prepared statements

## Structure of the program:

### Algorithm Overview (pictures of code are added for the more unique algorithms):

Class: AustinsClosetMySQL.java	
Algorithm:	Purpose:
1. <code>ArrayList&lt;SearchCloset&gt;</code>	<p>Used in the search <i>methods</i> in order to find clothes within the database and return items which match the user input.</p> <pre data-bbox="537 562 1414 659">public ArrayList&lt;SearchCloset&gt; shortSearch() {     ArrayList&lt;SearchCloset&gt; searchShorts = new ArrayList&lt;SearchCloset&gt;(); }</pre>
2. Various 'clothes' Search methods	<p>Use SearchCloset ArrayList to find items using <i>if try and catch statements</i> to match user specifications and <i>math.random</i> for colors. Also makes use of <i>while loops</i> to select 2 options of clothing.</p> <pre data-bbox="537 856 1484 1703">public ArrayList&lt;SearchCloset&gt; pantsSearch() {     ArrayList&lt;SearchCloset&gt; searchPants = new ArrayList&lt;SearchCloset&gt;();      Statement st;     ResultSet rs;      boolean shirt = false;     boolean pants = false;     boolean shorts = false;     boolean coats = false;     boolean accessory = false;      String material = "";     String shirtColor = "";     String pantsColor = "";     String shortsColor = "";     String coatsColor = "";     String accessoryColor = "";      var temp = tempDDwn.getSelectedItem().toString();     var weather = weatherDDwn.getSelectedItem().toString();     var occassion = occassionDDwn.getSelectedItem().toString();     var actLevel = actLevelDDwn.getSelectedItem().toString();      if(temp == "55-65"    temp == "45-55"    temp == "32-45"    temp == "&lt;32"){         shorts = true;     }     if(temp == "65-75"    temp == "75-85"    temp == "85-95"    temp == "&gt;95"){         pants = true;         coats = true;     }     //etc. below...</pre>

<p>3. <b>Various show‘Clothes’ methods</b></p>	<p>Uses <i>inheritance</i>, takes what was found in the ‘clothesSearch’ methods to display in the JTable of NetBeans.</p> <pre data-bbox="545 321 1468 590"> public void showShorts(){     ArrayList&lt;SearchCloset&gt; list = shortSearch();     DefaultTableModel model = (DefaultTableModel)OutfitOutput_Table.getModel();     Object[] row = new Object[3];     for(int i = 0; i &lt;list.size(); i++){         row[0] = list.get(i).getName();         row[1] = list.get(i).getType();          model.addRow(row);     } } </pre>
<p>4. <b>Update Database method</b></p>	<p>Uses a <i>MySQL connection</i> to update the JTable of Netbeans from the database in MySQL.</p>
<p>5. <b>Exit Button method</b></p>	<p>Uses <i>if statements</i> to determine if the user would like to exit the program.</p>
<p>6. <b>Reset Button method</b></p>	<p>Sets user input fields back to default.</p>
<p>7. <b>Add Button method</b></p>	<p>Uses <i>MySQL connection</i> and <i>prepared statements</i>, as well as try and <i>catch statements</i> to add items in the NetBeans JTable and the MySQL database under its inputted categories.</p> <pre data-bbox="532 1098 1471 1633"> private void addBtnActionPerformed(java.awt.event.ActionEvent evt) {     //add button uses mySQL and Netbeans to upload to DB and also to JTable     try{         Class.forName("com.mysql.cj.jdbc.Driver");         sqlConn = DriverManager.getConnection(dataConn, username, password);         pst = sqlConn.prepareStatement("insert into whole_closet(Type, Color, Pattern,"             + " Material, Formality, Activity, Name) VALUES (?, ?, ?, ?, ?, ?, ?)");          pst.setString(1, typeDDwn.getSelectedItem().toString());         pst.setString(2, colorDDwn.getSelectedItem().toString());         pst.setBoolean(3, patternCheckBox.isSelected());         pst.setString(4, materialDDwn.getSelectedItem().toString());         pst.setString(5, formalityDDwn.getSelectedItem().toString());         pst.setString(6, activityLevelDDwn.getSelectedItem().toString());         pst.setString(7, newClothesNameInput.getText());          pst.executeUpdate();         JOptionPane.showMessageDialog(this, "New item added");         upDateDB();     }     catch (Exception ex){         JOptionPane.showMessageDialog(null, ex);     } } </pre>

<p>8. <b>Select item method</b></p>	<p>When you click an item in JTable, the program will select it within MySQL for other purposes (such as deleting).</p> <pre data-bbox="529 279 1495 583"> private void netBeansWholeClosetTblMouseClicked(java.awt.event.MouseEvent evt) { //this is when you click the table to delete, it notes what you select and finds it in DB //to delete it when you run the deletebtn method DefaultTableModel RecordTable = (DefaultTableModel)netBeansWholeClosetTbl.getModel(); int SelectedRows = netBeansWholeClosetTbl.getSelectedRow();  typeDDwn.setSelectedItem(RecordTable.getValueAt(SelectedRows, 1).toString()); colorDDwn.setSelectedItem(RecordTable.getValueAt(SelectedRows, 2).toString()); patternCheckBox.setText(RecordTable.getValueAt(SelectedRows, 3).toString()); materialDDwn.setSelectedItem(RecordTable.getValueAt(SelectedRows, 4).toString()); formalityDDwn.setSelectedItem(RecordTable.getValueAt(SelectedRows, 5).toString()); activityLevelDDwn.setSelectedItem(RecordTable.getValueAt(SelectedRows, 6).toString()); newClothesNameInput.setText(RecordTable.getValueAt(SelectedRows, 7).toString()); } </pre>
<p>9. <b>Delete Button method</b></p>	<p>Uses <i>if, try and catch statements</i> in order to delete selected items from within the NetBeans JTable and its corresponding position within the MySQL database.</p> <pre data-bbox="529 762 1495 1413"> private void deleteBtnActionPerformed(java.awt.event.ActionEvent evt) { //this is for deleting from the table in NB and in MYSQL DefaultTableModel RecordTable = (DefaultTableModel)netBeansWholeClosetTbl.getModel(); int SelectedRows = netBeansWholeClosetTbl.getSelectedRow();  try{ id = Integer.parseInt(RecordTable.getValueAt(SelectedRows, 0).toString());  deleteItem = JOptionPane.showConfirmDialog(null, "Confirm if you want to delete this item.", "Warning", JOptionPane.YES_NO_OPTION); if(deleteItem == JOptionPane.YES_OPTION){ //gives warning to make sure user does want to delete  Class.forName("com.mysql.cj.jdbc.Driver"); sqlConn = DriverManager.getConnection(dataConn, username, password); pst = sqlConn.prepareStatement("Delete from whole_closet where id =?"); //mysql language in orange "" to execute command in mysql  pst.setInt(1, id); pst.executeUpdate(); JOptionPane.showMessageDialog(this, "Closet updated, item has been deleted"); updateDB();  typeDDwn.setSelectedItem(""); colorDDwn.setSelectedItem(""); patternCheckBox.setText(""); materialDDwn.setSelectedItem(""); formalityDDwn.setSelectedItem(""); activityLevelDDwn.setSelectedItem(""); newClothesNameInput.setText(""); } } catch(ClassNotFoundException ex){ java.util.logging.Logger.getLogger(AustinsClosetMySQL.class.getName()).log(java.util.logging.Level.SEVERE, null, ex); } catch(SQLException ex){ System.err.println(ex); } } } </pre>
<p>10. <b>Create Button method</b></p>	<p>When clicked, calls the show'Clothes' methods above to output 2 outfit options in the NetBeans JTable.</p>

<b>Class: SearchCloset</b>	
<b>Algorithm:</b>	<b>Purpose:</b>
1. SearchCloset	<p>Makes <i>objects</i> from the user input to for use in the <i>ArrayList</i> to search the clients closet and output outfits.</p> <pre>public class SearchCloset {     private int ID;     private String Name;     private String Type;     private String Color;     private String Pattern;     private String Material;     private String Formality;     private String Activity;      public SearchCloset(int ID, String Name, String Type, String Color, String Pattern,                         String Material, String Formality, String Activity)     {         this.ID = ID;         this.Name = Name;         this.Type = Type;         this.Color = Color;         this.Pattern = Pattern;         this.Material = Material;         this.Formality = Formality;         this.Activity = Activity;     }      public int getID()     {         return ID;     }     public String getName()     {         return Name;     } }</pre>

### Input/output:

There are 2 input fields in this program. One field for adding clothes to my clients closet based on the categories chosen by the user, and the other for determining the output of the outfit options.

For adding clothing, the user can pick categories from:	For determining the outfit output (outfit creation):
Type	Temperature
Color	Weather
Activity Level	Occasion
Formality	Activity Level
Material	
Pattern? Y/N	

### Why OOP and program structure?:

I used OOP as I could have multiple instances of the same kind of object. I was also able to use *inheritance* from my main class to my subclass for reusability purposes as there were a lot of similar methods when obtaining clothing items. OOP also allowed for a level of *abstraction*, which made it easy for me to isolate problems and debug, and the modularity that abstraction offers allows me to be able to easily reuse some of my code or tweak it for improvements.

### Data Structures Used:

#### ArrayList

- Used to dynamically store the *object elements* created for this program, and it makes manipulation of these objects easy through other methods. Therefore, *ArrayList* was the best option for the handling and procurement of the clothing items in this program.

#### User Defined Objects

- *Objects* were used as each item of clothing had attributes which were unique to it and this allowed also for *interoperability* from NetBeans into MySQL where the closest data is managed and stored.

## User Interface/ GUI work:

In this program, Java Swing components such as combo boxes, text boxes, tables and buttons were used to enhance functionality and ease of use for the client. The layout was all done in one tab, as a “one-stop-shop” for my client where everything was made to be easily accessible.

**Austin's Closet**

Type:

Color:

Material:

Formality:

Activity Level:

Pattern?

What would you like to call this item?

ID	Article Name
----	--------------

**Create Outfit:**

What is today's average temperature?

What is the weather today?

What occasion are you planning for?

What is your level of activity today?

**Here are your 2 outfit options:**

Name	Type
------	------

## Software tools used:

The software used for this IA was NetBeans, which is an Integrated Development Environment that offers software development along with a comprehensive GUI in order to create a finished, user-ready project. MySQL and MySQL Workbench, which are open source database management systems, were also used as a database to store and manage my client’s data about their clothing.