**PROJECT Java GUI Application: Java Jim’s Coffee Shoppe**

**Objective** To type a simple Java program, execute ( run ) the program for some particular values, observe the output and then modify the program.

***PROJECT DESCRIPTION***

Design an application that uses Java Graphical User Interface ( GUI ) components such as frames, text boxes, labels, checkboxes, radio buttons and click buttons.

The initial starter code for your program is given in **Figure 1** .

First run and test the starter code, which primarily has Java GUI elements and has you entering your name into a textbox ( input ) , selecting a radio button and selecting none, one or more checkboxes and then clicking a button to observe the output that appears within another textbox ( output ) .

After you test the original starter code and verify its functionality, you will then modify the program according to the instructions that follow.

Your completed program ( after modification ) will perform, at the minimum, each of the following tasks

• allow the user to enter their name in a text field element

• allow the user to select a coffee / beverage size, whose prices are given as:

( small : $ 1.25 ; medium : $ 1.75 ; large : $ 2.30 )

• allow the user to select none, one or more of: cream, raw sugar, espresso shot

( espresso shot price : $ 0.75 )

• when the [ COMPUTE ] button is clicked a text field will display a message to thank the program user and a message box displays a summary of the user’s coffee order

• when the [ EXIT ] button is clicked the application will close

Basically, your program is outlined in the given starter code statements shown within **Figure 1** , which follows. Review the starter code to understand the mechanisms of the interactions between the two classes. Perform any modifications according to this project’s instructions.

Type, compile and run the basic Java program that is shown in **Figure 1** , which follows. Then modify the program accordingly.

***Information About This Project***

GUI objects in Java include these elements.

frames and containers buttons

check boxes labels

radio buttons textboxes

***Steps to Complete This Project***

**STEP 1**  **Open NetBeans**

Open NetBeans and create a Java project with the following details.

For Project Name include **Lab12**

For the Main Class include **lab12.CoffeeMachine**

In your **Code** window for this class, shown below, copy the program code shown in **Figure 1** below, in the appropriate places, except substitute your own name in place of Sammy Student.

**PROJECT Java GUI Application: Java Jim’s Coffee Shoppe**

**Figure 1 Source Code for the Coffee Machine GUI Program**

|  |
| --- |
| **import java.awt.\*;**  **import java.awt.event.\*;**  **import javax.swing.\*;**  **//Sammy Student**  **public class CofeeMachine extends JFrame implements ActionListener**  **{**  **private static final long *serialVersionUID* = 1L;**  **JLabel l1, l2, l3, l4, l5, l6;**  **JButton b1, b2;**  **JTextField t1, t2, t3;**  **JCheckBox cream, raw, espresso;**  **private JRadioButton small;**  **private JRadioButton large;**  **private ButtonGroup group;**  **CofeeMachine()**  **{**  **l1 = new JLabel(" Customer Name");**  **l2 = new JLabel(" amount to pay");**  **l3 = new JLabel(" ");**  **l4 = new JLabel(" ");**  **l5 = new JLabel(" ");**  **l6 = new JLabel(" ");**  **b1 = new JButton("COMPUTE");**  **b2 = new JButton("EXIT");**    **t1 = new JTextField(10);**  **t2 = new JTextField(10);**  **small = new JRadioButton("small", true);**  **large = new JRadioButton("large", false);**  **group = new ButtonGroup();**  **group.add(small);**  **group.add(large);**  **cream = new JCheckBox("cream", false);**  **raw = new JCheckBox("raw sugar", false);**  **espresso = new JCheckBox("espresso shot", false);** |

**PROJECT Java GUI Application: Java Jim’s Coffee Shoppe**

**Figure 1 Source Code for the Coffee Machine GUI Program ( continued )**

|  |
| --- |
| **add(l1);**  **add(t1);**  **add(small);**  **add(cream);**  **add(large);**  **add(raw);**  **add(l3);**  **add(espresso);**  **add(l2);**  **add(t2);**  **add(l5);**  **add(l6);**  **add(b1);**  **add(b2);**  **b1.addActionListener(this);**  **b2.addActionListener(e -> System.*exit*(0));**  **setSize(500,300);**  **setLayout(new GridLayout(7,2));**  **setTitle("Coffee Machine");**  **}**  **public void actionPerformed(ActionEvent ae)**  **{**  **float price = 0;**  **String a = "", message = "";**  **if (cream.isSelected() == true)**  **{**  **// perform a task ...**  **}**  **if (raw.isSelected() == true)**  **{**  **// perform a task ...**  **}**  **if (espresso.isSelected() == true)**  **{**  **// perform a task ...**  **}**  **if (small.isSelected() == true)**  **{**  **System.*out*.println("small");**  **// perform a task ...**  **}**  **if (large.isSelected() == true)**  **{**  **System.*out*.println("large");**  **// perform a task ...**  **}** |

**PROJECT Java GUI Application: Java Jim’s Coffee Shoppe**

**Figure 1 Source Code for the Coffee Machine GUI Program ( continued )**

|  |
| --- |
| **if(ae.getSource() == b1)**  **{**  **a = t1.getText();**  **message = "your order: " + a + " $" + price;**  **t2.setText("thank you: " + a );**  **}**  **JOptionPane.*showMessageDialog*(null, "Summary: \n" + message,**  **"Order Summary", JOptionPane.*PLAIN\_MESSAGE*);**  **}**  **public static void main(String args[])**  **{**  **CofeeMachine a = new CofeeMachine();**  **a.setVisible(true);**  **a.setLocation(200,200);**  **}**  **}** |

**STEP 2 Build, Compile and Run the Program**

From the NetBeans Run menu select Run Project ( Lab12 ) to run your app.

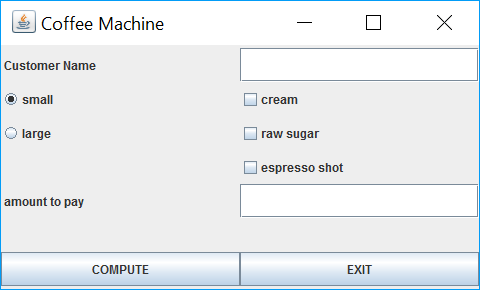
**STEP 3 Test the Program**

Once you have successfully compiled your program, review the graphical user interface that appears, as shown in **Figure 2** . This represents a sample run of the program.

**Figure 2 Java Swing Interface**

With the program running enter a valid input in the top text box that appears.

Click the button and observe the output. Specifically, with your application running you can enter your name as the customer, select a coffee cup size and choose to have cream, raw sugar and / or an espresso shot added to your coffee beverage. When you click the [ COMPUTE ] button an output will appear in the text field next to the [ amount to pay ] label and an initial summary of your coffee beverage order will appear in a message box.

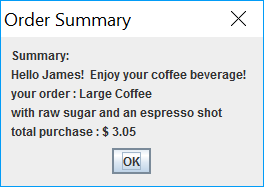


**PROJECT Java GUI Application: Java Jim’s Coffee Shoppe**

**STEP 4 Modify the Program**

With your initial starter program tested and functioning properly, you will now alter the program such that when the user clicks the [ COMPUTE ] button

The amount to pay will appear in the output text field and an order summary will appear in the message box, similar to that shown below.



**STEP 5 Modify Again the Program**

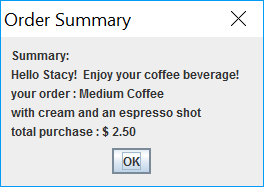
You will now expand your program’s layout by including a new row of GUI elements.

To accomplish this, perform these tasks:

• Add a new radio button to the Radio Button Group. This button will allow the program user to select a medium coffee beverage. The users may now be able to select from either a small, medium or large cup of coffee.

• In the same GUI row, add a label that will be used as a spacer element. Set the text property of the label to an empty string.

Save your program and perform a trial run of it. Test your program with data similar to that shown below.



**PROJECT Java GUI Application: Java Jim’s Coffee Shoppe**

Extra Credit: Add some exception handling to your code. Error trap any numeric characters input into textfield t1 by the user and allow for reinput if that is the case. Also add in an animated gif at the bottom area of your app, perhaps a cup of freshly brewed steaming coffee!

**STEP 6 Submit Your Project**

Once you have determined that your modified program is correctly displaying the required information, complete the submission process as follows:

Open MS Word and type a heading for a new document that includes your full name, course number, lab number and date.

Within the document paste snapshots of your modified program in action. Label the snapshots of your modified run with a reasonable description.

After your snapshot, paste in your finished source code as well copied in from your NetBeans editor.

Submit your MS Word document to Blackboard when complete.