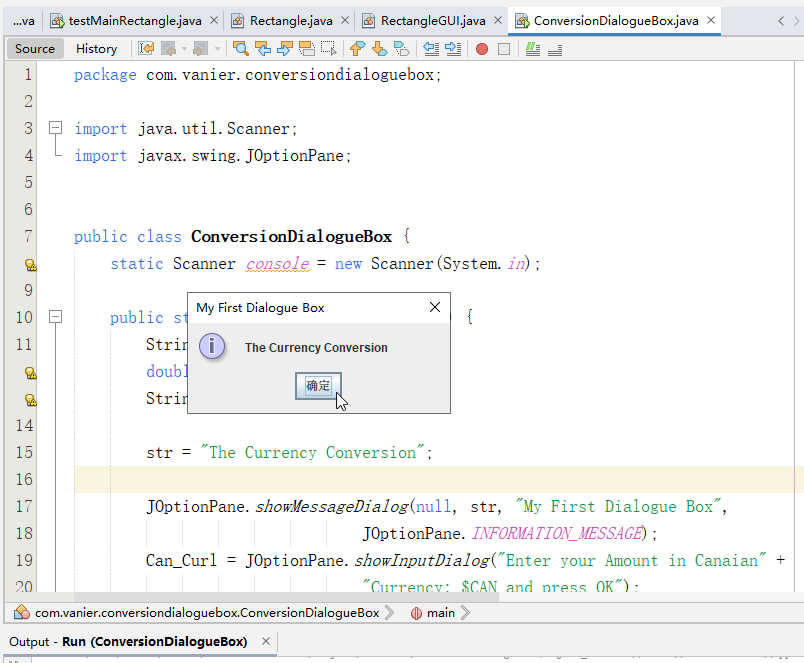
Lab 4: Advanced Graphical User Interface using Java

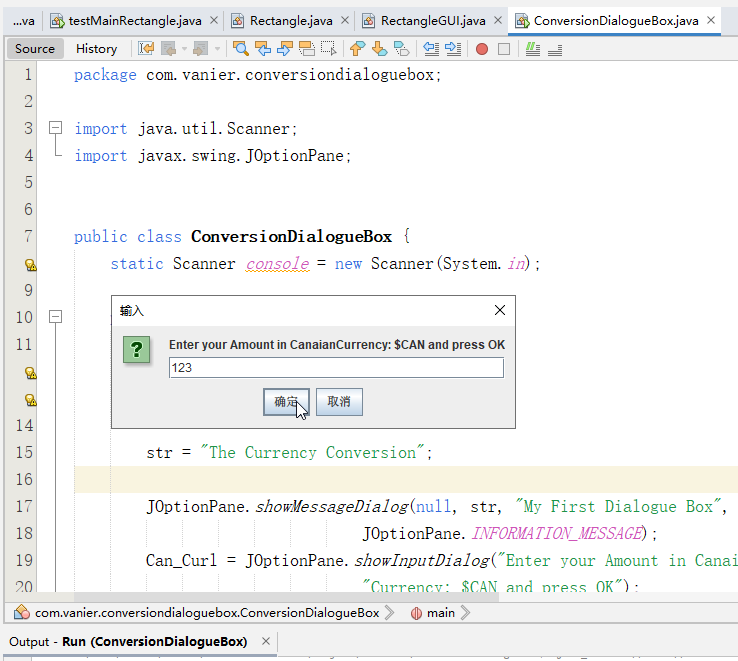
1. Graphical User Interface

a) Dialogue Boxes

Create the following Java program that display dialog boxes. These methods are contained in the class

JOptionPane and this class is contained in the package javax.swing



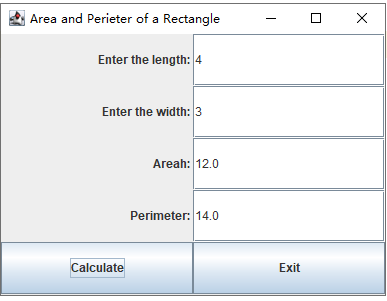


b) The class JFrame, JLabel, JTextField, JButton

The Graphical user interface (GUI) may contain Button objects from JButton class, label objects from

JLabel class, text field objects from JTextField class. All these objects are added within frame from

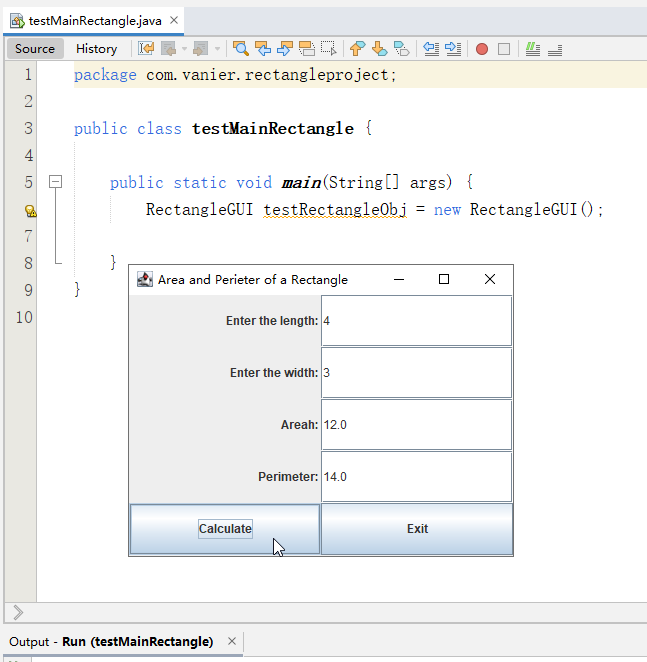
JFrame class. Create a Project to be named RectangleProject.



c) Create the Project RectangleProject.

Complete all these following programs as explained in my Lab 4 YouTube Video 1. Notice all missing coding statements are presented in this video with explanation.

1. **package** com.vanier.rectangleproject;
2. **import** javax.swing.\*;
3. **import** java.awt.\*;
4. **import** java.awt.event.\*;
6. **public** **class** RectangleGUI **extends** JFrame **implements** ActionListener {
7. //Create the four text fields
8. JTextField lengthTF, widthTF, areaTF, perimeterTF;
10. **public** RectangleGUI() {
11. //Create the four labels
12. JLabel lengthL = **new** JLabel("Enter the length: ", SwingConstants.RIGHT);
13. JLabel widthL = **new** JLabel("Enter the width: ", SwingConstants.RIGHT);
14. JLabel areaL = **new** JLabel("Areah: ", SwingConstants.RIGHT);
15. JLabel perimeterL = **new** JLabel("Perimeter: ", SwingConstants.RIGHT);
17. //Create the four text fields
18. lengthTF = **new** JTextField(10);
19. widthTF = **new** JTextField(10);
20. areaTF = **new** JTextField(10);
21. perimeterTF = **new** JTextField(10);
23. //Create Calculate Button
24. JButton calculateB = **new** JButton("Calculate");
26. //Associate or register this listener with the corresponding JButton
27. calculateB.addActionListener(**this**);
29. //Create Exit Buttonthis
30. JButton exitB = **new** JButton("Exit");
31. //Associate or register this listener with the corresponding JButton
32. exitB.addActionListener(**this**);
34. //Set the title of the window
35. setTitle("Area and Perieter of a Rectangle");
37. //Get the container
38. Container pane = getContentPane();
40. //Set the layout
41. pane.setLayout(**new** GridLayout(5, 2));
43. //Place the components in the pane
44. pane.add(lengthL);
45. pane.add(lengthTF);
46. pane.add(widthL);
47. pane.add(widthTF);
48. pane.add(areaL);
49. pane.add(areaTF);
50. pane.add(perimeterL);
51. pane.add(perimeterTF);
52. pane.add(calculateB);
53. pane.add(exitB);
55. //Set the size of the window and display it
56. setSize(400, 300);
57. setVisible(**true**);
58. } // end of constructor RectangleGUI
60. **public** **void** actionPerformed(ActionEvent e) {
61. **if**(e.getActionCommand().equals("Calculate")) {
62. **double** widthInput, lengthInput, areaOutput, perimeterOutput;
64. Rectangle myrectangle1 = **new** Rectangle();
66. lengthInput = Double.parseDouble(lengthTF.getText());
67. widthInput = Double.parseDouble(widthTF.getText());
69. myrectangle1.setDimension(lengthInput, widthInput);
71. areaOutput = myrectangle1.area();
72. perimeterOutput = myrectangle1.perimeter();
74. areaTF.setText(areaOutput + "");
75. perimeterTF.setText(perimeterOutput + "");
77. } **else** **if**(e.getActionCommand().equals("Exit")) {
78. System.exit(0);
79. }  //End of ActionPerformed interface Method
80. }
81. }



2. Sport Training Application with Graphical User Interface:

Create a Java Project to be named **GUISportProject** using NetBeans IDE which includes GUI

Java components that allows end user to evaluate the cost of a given sport training. The user

has to enter the variables **u\_Name**, u\_**Number\_hour, u\_Number\_week, u\_CostHour** through

JTextField.

The program will display the cost of sport training within an appropriate JTextField.

a) You need to design a **Java class** called **Sport (The same as Lab 1 Advanced Java)**, which

takes Name, Number of hours per week, Number of weeks as three **private non static**

members called respectively **name, number\_hour, number\_week**. A variable hourly rate of sport

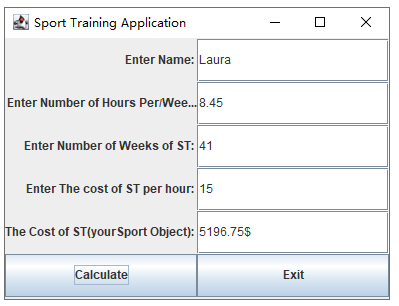
training **cost\_hour** as **public static** data member.

b) Create SportGInterface.java to describe the User Graphical Interface that includes the different

Graphical components displayed in Figure 1 (5 labels, 5 Text Fields, and 2 buttons).

c) Create TestSportTrainingProgram.java to test SportGInterface class, where you instantiate the

object of SportGInterface class type to display the Figure 1.



SportGInterface.java

1. **package** com.vanier.guisportproject;
2. **import** javax.swing.\*;
3. **import** java.awt.\*;
4. **import** java.awt.event.\*;
6. **public** **class** SportGInterface **extends** JFrame **implements** ActionListener {
7. //Create the four text fields
8. JTextField u\_NameTF, u\_Number\_hourTF, u\_Number\_weekTF, u\_CostHourTF, u\_CostTF;
10. **public** SportGInterface() {
11. //Create the four labels
12. JLabel u\_Name = **new** JLabel("Enter Name: ", SwingConstants.RIGHT);
13. JLabel u\_Number\_hour = **new** JLabel("Enter Number of Hours Per/Week: ", SwingConstants.RIGHT);
14. JLabel u\_Number\_week = **new** JLabel("Enter Number of Weeks of ST: ", SwingConstants.RIGHT);
15. JLabel u\_CostHour = **new** JLabel("Enter The cost of ST per hour: ", SwingConstants.RIGHT);
16. JLabel u\_Cost = **new** JLabel("The Cost of ST(yourSport Object): ", SwingConstants.RIGHT);
17. //Create the four text fields
18. u\_NameTF = **new** JTextField(10);
19. u\_Number\_hourTF = **new** JTextField(10);
20. u\_Number\_weekTF = **new** JTextField(10);
21. u\_CostHourTF = **new** JTextField(10);
22. u\_CostTF = **new** JTextField(10);
24. //Create Calculate Button
25. JButton calculateB = **new** JButton("Calculate");
27. //Associate or register this listener with the corresponding JButton
28. calculateB.addActionListener(**this**);
30. //Create Exit Buttonthis
31. JButton exitB = **new** JButton("Exit");
32. //Associate or register this listener with the corresponding JButton
33. exitB.addActionListener(**this**);
35. //Set the title of the window
36. setTitle("Sport Training Application");
38. //Get the container
39. Container pane = getContentPane();
40. //Set the layout
41. pane.setLayout(**new** GridLayout(6, 2));
43. //Place the components in the pane
44. pane.add(u\_Name);
45. pane.add(u\_NameTF);
46. pane.add(u\_Number\_hour);
47. pane.add(u\_Number\_hourTF);
48. pane.add(u\_Number\_week);
49. pane.add(u\_Number\_weekTF);
50. pane.add(u\_CostHour);
51. pane.add(u\_CostHourTF);
52. pane.add(u\_Cost);
53. pane.add(u\_CostTF);
54. pane.add(calculateB);
55. pane.add(exitB);
57. //Set the size of the window and display it
58. setSize(400, 300);
59. setVisible(**true**);
60. }
62. **public** **void** actionPerformed(ActionEvent e) {
63. **if**(e.getActionCommand().equals("Calculate")) {
64. String u\_NameInput;
65. **int** u\_Number\_weekInput;
66. **double** u\_Number\_hourInput, u\_CostHourInput;
67. **double** u\_CostOutput;
69. u\_NameInput = u\_NameTF.getText();
70. u\_Number\_hourInput = Double.parseDouble(u\_Number\_hourTF.getText());
71. u\_Number\_weekInput = Integer.parseInt(u\_Number\_weekTF.getText());
72. u\_CostHourInput = Double.parseDouble(u\_CostHourTF.getText());
73. Sport mysport1 = **new** Sport(u\_NameInput,u\_Number\_hourInput,u\_Number\_weekInput);
75. u\_CostOutput = mysport1.CalculateCostTraining(u\_CostHourInput);
76. u\_CostTF.setText(String.format("%.2f", u\_CostOutput).toString() + "$");
77. } **else** **if**(e.getActionCommand().equals("Exit")) {
78. System.exit(0);
79. }  //End of ActionPerformed interface Method
80. }
81. }

**3. Review Questions**

A. Write necessary statements to create the following GUI components:

a. A JLabel with the text string "Enter the number of courses".

JLabel num\_courses = new JLabel("Enter the number of courses: ", SwingConstants.RIGHT);

b. A JButton with the text string "Run".

JButton runB = new JButton("Run");

c. A JTextField that can display 15 characters.

textTF = new JTextField(15);

d. A window with the title "Welcome Home!".

//Set the title of the window

setTitle("Welcome Home!");

//Get the container

Container pane = getContentPane();

e. A grid layout object of 5 rows and 4 columns.

GridLayout newGridLayout = new GridLayout(5, 4));

f. A JButton that register a listener.

runB.addActionListener(this);

g. A JButton object added to a container.

pane.add(runB);

h. A container layout set to grid layout.

pane.setLayout(newGridLayout);

B. True or False and Why

a. actionPerformed is an abstract method interface defined in interface actionListener

True

The actionListener is an interface (not a class) that contains a single method: public void actionPerformed(ActionEvent evt) , and a class that implements the interface must contain an actionPerformed() method.

b. no need to register a listener to a button.

False

The program must register this object as an action listener on the button (the event source), using the addActionListener method.

c. Container is a space within the frame.

False

Containers are an integral part of SWING GUI components. A container provides a space where a component can be located. A Container in AWT is a component itself and it provides the capability to add a component

Frame which a JFrame is a top-level window with a title and a border.

JFrame is a heavy weight container used as the top-level window. JPanel is a light weight container used to organize GUI components.

d. getContentPane() is a method of Container class

False

JFrame provides two methods: getContentPane and setContentPane are used to get and set its Content Pane, getContentPane() is a method of JFrame class

e. You can add the control elements to JFrame object

True

JFrame frame = new JFrame("This is a Frame");

frame.add(new JButton("This is a Button"));

f. addActionListener method allows you to register a listener to JTextField

True

JTextField textField = new JTextField();

add(textField);

textFiledActionListener textFiledActionListener = new textFiledActionListener();

textField.addActionListener(textFiledActionListener);

g. addActionListener method allows you to register a listener to JButton

True

JButton button = new JButton("buttom");

ActionListener actionListener = new MyActionListener();

button.addActionListener(actionListener);

h. You add control elements within a frame set to grid layout

True

JFrame frame = new JFrame("This is a Frame");

GridLayout gridLayout = new GridLayout(1,1);

frame.setLayout(gridLayout);

frame.add(new JButton("This is a Button"));

C. Multiple choice (only one answer per question is valid)

A 1. The method of the class JFrame used to access the content pane of the window.

a. getContentPane b. setContentPane

c. getPane d. getContent

C 2. Which method contains the code that the program executes when a specific event is generated?

a. buttonListener c. actionPerformed

b. GUIListener d. windowListener

B 3. Suppose that you create an application in which you instantiate a JFrame named frame1 and a JLabel named label1. Which of the following statements within the application adds label1 to frame1?

a. label1.add(frame1); b. frame1.add(label1);

c. this.add(label1); d. two of the above

A 4. Event listeners must

a. implement an interface

b. be included in private inner classes

c. not receive any arguments

d. exit the application once it has handled the event

C 5. A method that register a listener to button is a \_\_\_\_\_

a. add b. setLayout

c. addActionListener d. setText

D 6. Which of the following methods is part of the class JTextField?

a. setVisible c. setTitle

b. setSize d. getText