****

**FACULTY OF COMPUTING AND INFORMATICS**

**HC00 SOFTWARE ENGINEERING**

**SEMESTER 2 ( 2019/2020)**

**KK14203 OBJECT ORIENTED PROGRAMMING**

**SECTION 1**

**PROJECT 2 ( INDIVIDUAL ASSIGNMENT )**

**FATIN NUR LIANA BINTI MUSTAFFA KAMAL**

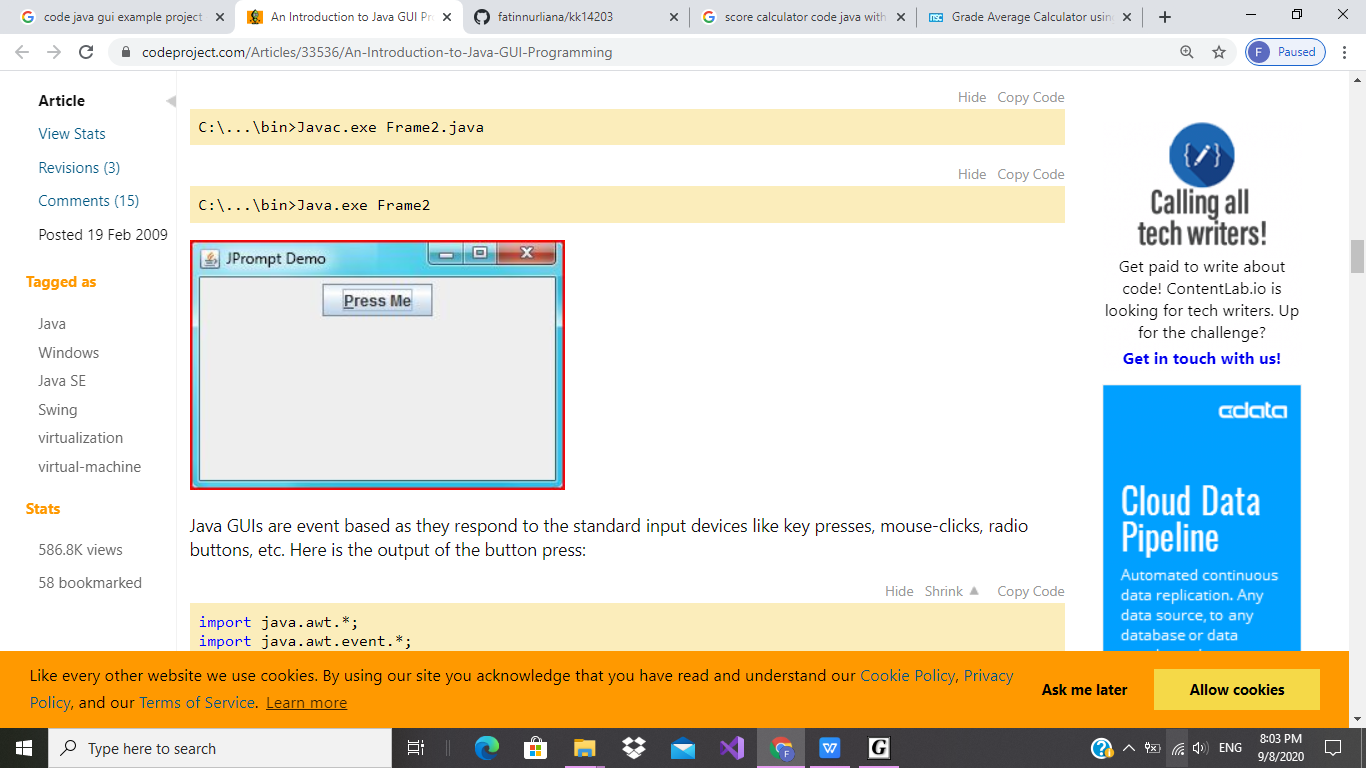
**DR. MOHD SHAMRIE SAININ**

**JAVA CODE**

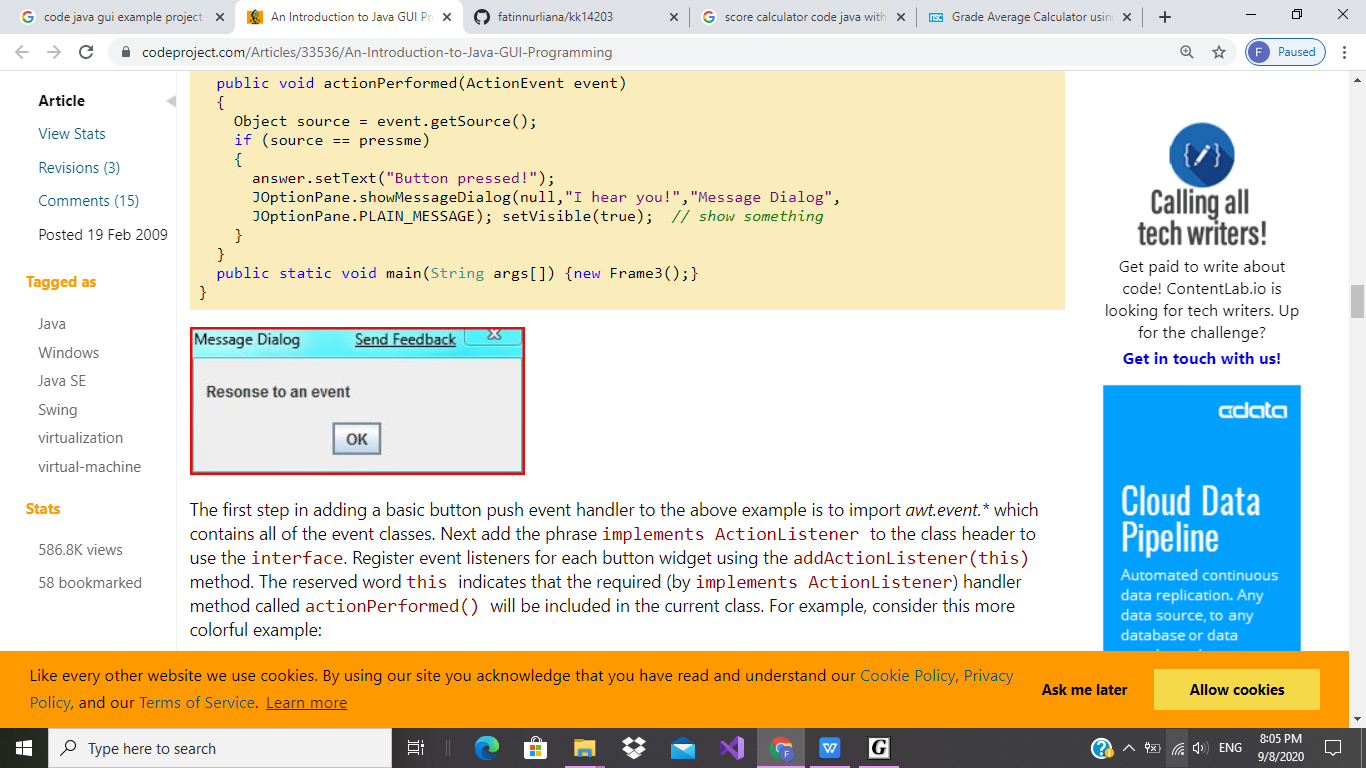
import java.awt.\*;  
import java.awt.event.\*;  
import javax.swing.\*; //notice javax  
public class Frame1 extends JFrame  
{  
 JPanel pane = new JPanel();  
 Frame1() // the frame constructor method  
 {  
 super("My Simple Frame"); setBounds(100,100,300,100);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 Container con = this.getContentPane(); // inherit main frame  
 con.add(pane); // add the panel to frame  
 // customize panel here  
 // pane.add(someWidget);  
 setVisible(true); // display this frame  
 }  
 public static void main(String args[]) {new Frame1();}  
}  
  
import java.awt.\*;   
import java.awt.event.\*;   
import javax.swing.\*;  
public class Frame2 extends JFrame  
{  
 JPanel pane = new JPanel();  
 JButton pressme = new JButton("Press Me");  
 Frame2() // the frame constructor  
 {  
 super("JPrompt Demo"); setBounds(100,100,300,200);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 Container con = this.getContentPane(); // inherit main frame  
 con.add(pane); // JPanel containers default to FlowLayout  
 pressme.setMnemonic('P'); // associate hotkey to button  
 pane.add(pressme); pressme.requestFocus();  
 setVisible(true); // make frame visible  
 }  
 public static void main(String args[]) {new Frame2();}  
}  
  
import java.awt.\*;   
import java.awt.event.\*;  
import javax.swing.\*;  
public class Frame3 extends JFrame implements ActionListener  
{  
 JLabel answer = new JLabel("");  
 JPanel pane = new JPanel(); // create pane object  
 JButton pressme = new JButton("Press Me");  
 Frame3() // the constructor  
 {  
 super("Event Handler Demo"); setBounds(100,100,300,200);  
 setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 Container con = this.getContentPane(); // inherit main frame  
 con.add(pane); pressme.setMnemonic('P'); // associate hotkey  
 pressme.addActionListener(this); // register button listener  
 pane.add(answer); pane.add(pressme); pressme.requestFocus();  
 setVisible(true); // make frame visible  
 }  
 // here is the basic event handler  
 public void actionPerformed(ActionEvent event)  
 {  
 Object source = event.getSource();  
 if (source == pressme)  
 {  
 answer.setText("Button pressed!");  
 JOptionPane.showMessageDialog(null,"I hear you!","Message Dialog",  
 JOptionPane.PLAIN\_MESSAGE); setVisible(true); // show something  
 }  
 }  
 public static void main(String args[]) {new Frame3();}  
}  
  
import javax.swing.\*;  
import java.awt.Color;  
  
public class ButtonDemo{  
  
 public JPanel createContentPane (){  
  
 // We create a bottom JPanel to place everything on.  
 JPanel totalGUI = new JPanel();  
 totalGUI.setLayout(null);  
  
 // Creation of a Panel to contain the title labels  
 JPanel titlePanel = new JPanel();  
 titlePanel.setLayout(null);  
 titlePanel.setLocation(10, 0);  
 titlePanel.setSize(250, 30);  
 totalGUI.add(titlePanel);  
  
 JLabel redLabel = new JLabel("Red Team");  
 redLabel.setLocation(0, 0);  
 redLabel.setSize(100, 30);  
 redLabel.setHorizontalAlignment(0);  
 redLabel.setForeground(Color.red);  
 titlePanel.add(redLabel);  
  
 JLabel blueLabel = new JLabel("Blue Team");  
 blueLabel.setLocation(120, 0);  
 blueLabel.setSize(100, 30);  
 blueLabel.setHorizontalAlignment(0);  
 blueLabel.setForeground(Color.blue);  
 titlePanel.add(blueLabel);  
  
 // Creation of a Panel to contain the score labels.  
 JPanel scorePanel = new JPanel();  
 scorePanel.setLayout(null);  
 scorePanel.setLocation(10, 40);  
 scorePanel.setSize(250, 30);  
 totalGUI.add(scorePanel);  
  
 JLabel redScore = new JLabel("0");  
 redScore.setLocation(0, 0);  
 redScore.setSize(100, 30);  
 redScore.setHorizontalAlignment(0);  
 scorePanel.add(redScore);  
  
 JLabel blueScore = new JLabel("0");  
 blueScore.setLocation(120, 0);  
 blueScore.setSize(100, 30);  
 blueScore.setHorizontalAlignment(0);  
 scorePanel.add(blueScore);  
  
 // Creation of a label to contain all the JButtons.  
 JPanel buttonPanel = new JPanel();  
 buttonPanel.setLayout(null);  
 buttonPanel.setLocation(10, 80);  
 buttonPanel.setSize(250, 70);  
 totalGUI.add(buttonPanel);  
  
 // We create a button and manipulate it using the syntax we have  
 // used before.  
 JButton redButton = new JButton("Red Score!");  
 redButton.setLocation(0, 0);  
 redButton.setSize(100, 30);  
 buttonPanel.add(redButton);  
  
 JButton blueButton = new JButton("Blue Score!");  
 blueButton.setLocation(120, 0);  
 blueButton.setSize(100, 30);  
 buttonPanel.add(blueButton);  
  
 JButton resetButton = new JButton("Reset Score");  
 resetButton.setLocation(0, 40);  
 resetButton.setSize(220, 30);  
 buttonPanel.add(resetButton);   
   
 totalGUI.setOpaque(true);  
 return totalGUI;  
 }  
  
 private static void createAndShowGUI() {  
  
 JFrame.setDefaultLookAndFeelDecorated(true);  
 JFrame frame = new JFrame("[=] JButton Scores! [=]");  
  
 //Create and set up the content pane.  
 ButtonExample demo = new ButtonExample();  
 frame.setContentPane(demo.createContentPane());  
  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 frame.setSize(250, 190);  
 frame.setVisible(true);  
 }  
  
 public static void main(String[] args) {  
 //Schedule a job for the event-dispatching thread:  
 //creating and showing this application's GUI.  
 SwingUtilities.invokeLater(new Runnable() {  
 public void run() {  
 createAndShowGUI();  
 }  
 });  
 }   
}  
  
  
import javax.swing.\*;  
import java.awt.Color;  
import java.awt.event.ActionListener;  
import java.awt.event.ActionEvent;  
  
public class ButtonDemo\_Extended implements ActionListener{  
  
 // Definition of global values and items that are part of the GUI.  
 int redScoreAmount = 0;  
 int blueScoreAmount = 0;  
  
 JPanel titlePanel, scorePanel, buttonPanel;  
 JLabel redLabel, blueLabel, redScore, blueScore;  
 JButton redButton, blueButton, resetButton;  
  
 public JPanel createContentPane (){  
  
 // We create a bottom JPanel to place everything on.  
 JPanel totalGUI = new JPanel();  
 totalGUI.setLayout(null);  
  
 // Creation of a Panel to contain the title labels  
 titlePanel = new JPanel();  
 titlePanel.setLayout(null);  
 titlePanel.setLocation(10, 0);  
 titlePanel.setSize(250, 30);  
 totalGUI.add(titlePanel);  
  
 redLabel = new JLabel("Red Team");  
 redLabel.setLocation(0, 0);  
 redLabel.setSize(120, 30);  
 redLabel.setHorizontalAlignment(0);  
 redLabel.setForeground(Color.red);  
 titlePanel.add(redLabel);  
  
 blueLabel = new JLabel("Blue Team");  
 blueLabel.setLocation(130, 0);  
 blueLabel.setSize(120, 30);  
 blueLabel.setHorizontalAlignment(0);  
 blueLabel.setForeground(Color.blue);  
 titlePanel.add(blueLabel);  
  
 // Creation of a Panel to contain the score labels.  
 scorePanel = new JPanel();  
 scorePanel.setLayout(null);  
 scorePanel.setLocation(10, 40);  
 scorePanel.setSize(260, 30);  
 totalGUI.add(scorePanel);  
  
 redScore = new JLabel(""+redScoreAmount);  
 redScore.setLocation(0, 0);  
 redScore.setSize(120, 30);  
 redScore.setHorizontalAlignment(0);  
 scorePanel.add(redScore);  
  
 blueScore = new JLabel(""+blueScoreAmount);  
 blueScore.setLocation(130, 0);  
 blueScore.setSize(120, 30);  
 blueScore.setHorizontalAlignment(0);  
 scorePanel.add(blueScore);  
  
 // Creation of a Panel to contain all the JButtons.  
 buttonPanel = new JPanel();  
 buttonPanel.setLayout(null);  
 buttonPanel.setLocation(10, 80);  
 buttonPanel.setSize(260, 70);  
 totalGUI.add(buttonPanel);  
  
 // We create a button and manipulate it using the syntax we have  
 // used before. Now each button has an ActionListener which posts   
 // its action out when the button is pressed.  
 redButton = new JButton("Red Score!");  
 redButton.setLocation(0, 0);  
 redButton.setSize(120, 30);  
 redButton.addActionListener(this);  
 buttonPanel.add(redButton);  
  
 blueButton = new JButton("Blue Score!");  
 blueButton.setLocation(130, 0);  
 blueButton.setSize(120, 30);  
 blueButton.addActionListener(this);  
 buttonPanel.add(blueButton);  
  
 resetButton = new JButton("Reset Score");  
 resetButton.setLocation(0, 40);  
 resetButton.setSize(250, 30);  
 resetButton.addActionListener(this);  
 buttonPanel.add(resetButton);  
   
 totalGUI.setOpaque(true);  
 return totalGUI;  
 }  
  
 // This is the new ActionPerformed Method.  
 // It catches any events with an ActionListener attached.  
 // Using an if statement, we can determine which button was pressed  
 // and change the appropriate values in our GUI.  
 public void actionPerformed(ActionEvent e) {  
 if(e.getSource() == redButton)  
 {  
 redScoreAmount = redScoreAmount + 1;  
 redScore.setText(""+redScoreAmount);  
 }  
 else if(e.getSource() == blueButton)  
 {  
 blueScoreAmount = blueScoreAmount + 1;  
 blueScore.setText(""+blueScoreAmount);  
 }  
 else if(e.getSource() == resetButton)  
 {  
 redScoreAmount = 0;  
 blueScoreAmount = 0;  
 redScore.setText(""+redScoreAmount);  
 blueScore.setText(""+blueScoreAmount);  
 }  
 }  
  
 private static void createAndShowGUI() {  
  
 JFrame.setDefaultLookAndFeelDecorated(true);  
 JFrame frame = new JFrame("[=] JButton Scores! [=]");  
  
 //Create and set up the content pane.  
 ButtonDemo\_Extended demo = new ButtonDemo\_Extended();  
 frame.setContentPane(demo.createContentPane());  
  
 frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);  
 frame.setSize(280, 190);  
 frame.setVisible(true);  
 }  
  
 public static void main(String[] args) {  
 //Schedule a job for the event-dispatching thread:  
 //creating and showing this application's GUI.  
 SwingUtilities.invokeLater(new Runnable() {  
 public void run() {  
 createAndShowGUI();  
 }  
 });  
 }  
}

**USER MANUAL**

1. **Interface score**



1. The first step in adding a basic button push event handler to the above



1. **OUTPUT**

