**CSC 1302: PRINCIPLES OF COMPUTER SCIENCE II**

**Lab 11**

**How to Submit**

Please submit your answers to the lab instructor once you have completed.

Failure to submit will result in a **ZERO FOR THIS LAB. NO EXCEPTIONS**.

Consider the following code:

// Coordinates. java

**import** javax.swing.JFrame;

**public** **class** Coordinates

{

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

{

JFrame frame = **new** JFrame("Coordinates");

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

frame.getContentPane().add(**new** CoordinatesPanel());

frame.pack();

frame.setVisible(**true**);

}

}

// CoordinatesPanel. java

**import** java.awt.\*;

**import** java.awt.event.\*;

**import** javax.swing.\*;

**public** **class** CoordinatesPanel **extends** JPanel

{

**private** **final** **int** SIZE = 6; // diameter of dot

**private** **int** x = 50, y = 50; // coordinates of mouse press

//-----------------------------------------------------------------

// Constructor: Sets up this panel to listen for mouse events.

//-----------------------------------------------------------------

**public** CoordinatesPanel()

{

addMouseListener(**new** CoordinatesListener());

setBackground(Color.***black***);

setPreferredSize(**new** Dimension(300, 200));

}

//-----------------------------------------------------------------

// Draws all of the dots stored in the list.

//-----------------------------------------------------------------

**public** **void** paintComponent(Graphics page)

{

**super**.paintComponent(page);

page.setColor(Color.***green***);

page.fillOval(x, y, SIZE, SIZE);

page.drawString("Coordinates: (" + x + ", " + y + ")", 5, 15);

}

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

// Represents the listener for mouse events.

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

**private** **class** CoordinatesListener **implements** MouseListener

{

//--------------------------------------------------------------

// Adds the current point to the list of points and redraws

// the panel whenever the mouse button is pressed.

//--------------------------------------------------------------

**public** **void** mousePressed(MouseEvent event)

{

x = event.getX();

y = event.getY();

repaint();

}

//--------------------------------------------------------------

// Provide empty definitions for unused event methods.

//--------------------------------------------------------------

**public** **void** mouseClicked(MouseEvent event) {}

**public** **void** mouseReleased(MouseEvent event) {}

**public** **void** mouseEntered(MouseEvent event) {}

**public** **void** mouseExited(MouseEvent event) {}

}

}

Update the following code so that:

1. When mouse is clicked, “Mouse clicked” text is printed;
2. When mouse is released, “Mouse released” text is printed;
3. When mouse enters the window, “Mouse entered” text is printed;
4. When mouse leaves the window, “Mouse exited” text is printed.

Hints:

* Use mouseClicked, mouseReleased, mouseEntered and mouseExited methods;
* Use a string field to store the current message (similar to the way coordinates are stored in the example). Use page.drawString method to print the current message.