

Eyasu Haile

CSC 133

Quiz #5

CODE:

```
package com.mycompany.a7;

import static com.codename1.ui.CN.*;
import com.codename1.ui.Display;
import com.codename1.ui.Form;
import com.codename1.ui.Button;
import com.codename1.ui.Component;
import com.codename1.ui.Container;
import com.codename1.ui.Dialog;
import com.codename1.ui.Label;
import com.codename1.ui.TextField;
import com.codename1.ui.plaf.UIManager;
import com.codename1.ui.util.Resources;
import com.codename1.charts.compat. .Style;
import com.codename1.io.Log;
import com.codename1.ui.Toolbar;
import com.codename1.ui.events.ActionEvent;
import com.codename1.ui.events.ActionListener;
import com.codename1.ui.geom.Dimension;

import java.io.IOException;

import com.codename1.ui.layouts.BorderLayout;
import com.codename1.ui.layouts.BoxLayout;
import com.codename1.ui.layouts.FlowLayout;
import com.codename1.ui.layouts.GridBagLayout;
import com.codename1.ui.layouts.Layout;
import com.codename1.io.NetworkEvent;

/**
 * This file was generated by <a href="https://www.codenameone.com/">Codename
 * One</a> for the purpose of building native mobile applications using Java.
 */
public class MyApplication {
    private Form current;
    private Resources theme;

    public void init(Object context) {
        // use two network threads instead of one
        updateNetworkThreadCount(2);
        theme = UIManager.initFirstTheme("/theme");
```

```

// Enable Toolbar on all Forms by default
Toolbar.setGlobalToolbar(true);

// Pro only feature, uncomment if you have a pro subscription
Log.bindCrashProtection(true);

addNetworkErrorListener(err -> {
    // prevent the event from propagating
    err.consume();
    if (err.getError() != null) {
        Log.e(err.getError());
    }
    Log.sendLogAsync();
    Dialog.show("Connection Error",
        "There was a networking error in the connection
to " + err.getConnectionRequest().getUrl(), "OK",
        null);
    });
}

public void start() {
    if (current != null) {
        current.show();
        return;
    }

    // This Form holds all components together
    Form hi = new Form("Form", new
BorderLayout(BorderLayout.CENTER_BEHAVIOR_CENTER));
    // Click only component
    Button b = new Button("West Region");
    b.getAllStyles().setFgColor(0xFF0000);
    // Release area of the Form
    Button y = new Button("RELEASE HERE ");
    y.getAllStyles().setFgColor(0x00A300);
    y.setDropTarget(true);
    // The component that is dragged and dropped
    Button x = new Button("X");
    x.getAllStyles().setFgColor(0x0000FF);
    x.setDraggable(true);

    b.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent ev) {
            // Clicked component prints msg
            System.out.println("Clicked From the West Region");
        }
    });

    x.addPointerPressedListener(new ActionListener() {

```

```

        public void actionPerformed(ActionEvent ev) {
            // Dragged from the center
            System.out.println("Dragged From Center Region");
            System.out.println("Pointer PRESSED X and Y: " +
x.getDraggedx() + ", " + x.getDraggedy());
        }
    });

    x.addPointerDraggedListener(new ActionListener() {
        public void actionPerformed(ActionEvent ev) {
            // System.out.println("CENTER");
            System.out.println("Pointer DRAGGED X and Y: " +
x.getDraggedx() + ", " + x.getDraggedy());
        }
    });

    x.addDropListener(new ActionListener() {
        public void actionPerformed(ActionEvent ev) {
            System.out.println("Pointer RELEASED X and Y: " +
x.getDraggedx() + ", " + x.getDraggedy());
        }
    });

    hi.add(BorderLayout.CENTER, x);
    hi.add(BorderLayout.EAST, y);
    hi.add(BorderLayout.WEST, b);
    hi.show();
}

public void stop() {
    current = getCurrentForm();
    if (current instanceof Dialog) {
        ((Dialog) current).dispose();
        current = getCurrentForm();
    }
}

public void destroy() {
}

}

```

OUTPUT:

