

SER516

Software Agility:

Project and Process Management

Lecture 06. Quiz 1

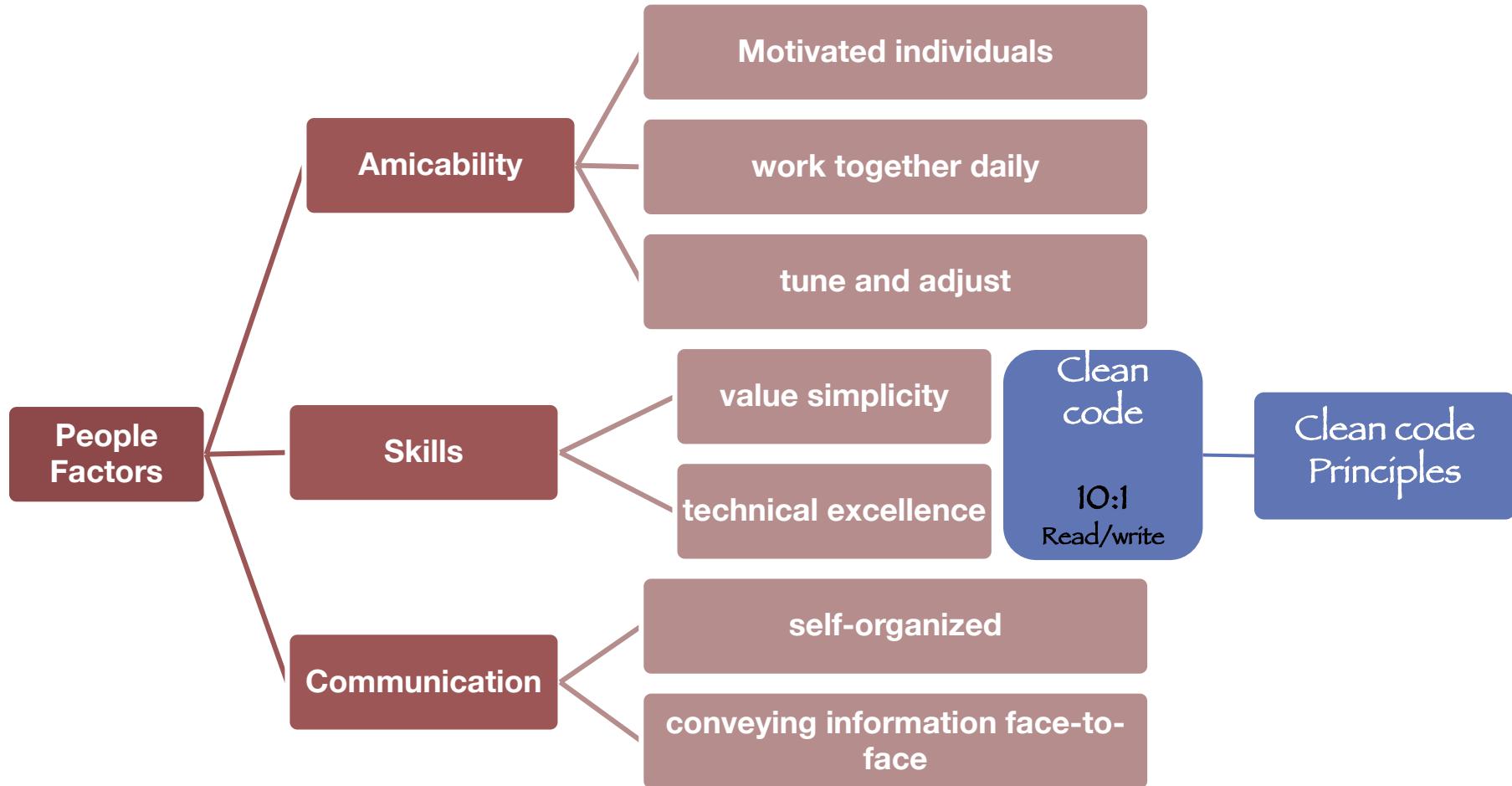
Javier Gonzalez-Sanchez

javiergs@asu.edu

javiergs.engineering.asu.edu

Office Hours: By appointment

Next Step



Does your project has clean code?

Avoid Code Bloat

Code

```
/*
 * SER 516 Lab 1
 * [REDACTED]
 * 01/22/2017
 *
 */

import java.awt.Color;
import java.awt.Component;
import java.awt.Font;

import javax.swingBoxLayout;
import javax.swing.JLabel;
import javax.swing.JPanel;

public class P013_[REDACTED]_Panel extends JPanel{

    /*This class is used to create a Panel of size 100X100 which contains three labels Firstname, Lastname
     *and number. This number increments from 0-9 if the value passed to the class is even and decrements
     *from 9-0 if the value passed is odd.
     */

    private JLabel firstname;
    private JLabel lastname;
    private JLabel number;
    private int flag;

    // This is the constructor which receives an integer value.
    public P013_[REDACTED]_Panel(int value){
        if(value%2 == 0)
            flag = 0; //set flag to 0 if value is even
        else
            flag = 1; //set flag to 1 if value is odd

        // Creation of JLabel
    }
}
```

Code

```
import javax.swing.Box;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 */
/*
 * @author [REDACTED]
 */
public final class P027_[REDACTED]_Panel extends JPanel{

    private Name name;
    private JLabel counter;
    boolean flag = false;
    int count=9;
```

```
/**
 * Row1 one is the first name.
 * Row 2 is the last name.
 * Row 3 is the counter that is incrementing from 0 or decrementing frrom 9.
 * The mode of the counter and the background color are determined by the value passed to the constructor.
 */
public class P016_[REDACTED]_Panel extends JPanel {
    private int timer;
    private int type;
    // private static JPanel jp;
    private Font f = new Font("Papyrus", Font.BOLD, 16);

    /**
     * constructor class
     * describes the background color and counter mode of the panel.
     */
    public P016_[REDACTED]_Panel(int type) { //constructor
        // Even is white and count up, odd is cyan and counts down.
        this.type = type;
        if (type % 2 == 0) {
            timer = 0;
        } else {
            timer = 9;
        }
        function(); // calling method which has the logic
    }
}
```

Code

```
import javax.swing.BorderFactory;
import javax.swing.BoxLayout;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.Timer;
import javax.swing.border.Border;

public class P005_[REDACTED]_Panel extends JPanel
{
    int count_even = 0;
    int flag;
    int count_odd = 9;

    //creating JPanel and JLabels for printing Firsname, Lastname and the timer field
    JPanel myPanel = new JPanel();
    private JLabel firstname = new JLabel("[REDACTED]", JLabel.CENTER);
    private JLabel lastname = new JLabel("[REDACTED]", JLabel.CENTER);
    private JLabel displayNumber = new JLabel(" 0 ", JLabel.CENTER);
    //private JLabel flagNum;
    Border border = BorderFactory.createLineBorder(Color.black, 1);

    //parameterized constructor to check if the input number is even or odd
    public P005_[REDACTED]_Panel(int number)
    {
        //JPanel objPanell = new JPanel();
        if ((number % 2) == 0)
        {
```

Code

```
75      //Add all tabs to tab pane
76      for(int i=0; i<TAB_NAMES.length; i++){
77
78          /* takes the class name as parameter and if the class
79          exists then adds its object to the passed tabbedPane */
80          try{
81              Class<?> clazz = Class.forName(TAB_NAMES[i]);
82              Method method = clazz.getMethod("getName");
83              Object newTab = clazz.getDeclaredConstructor().newInstance();
84              tabbedPane.addTab(method.invoke(newTab).toString(),
85                                (JPanel)newTab);
86          }
```

```
130
131      public static void main(String[] args) {
132
133          //schedules UI on separate thread as swing is not thread safe
134          javax.swing.SwingUtilities.invokeLater(new Runnable() {
135              public void run() {
136                  new Frame_81();
137              }
138          });
139      }
140  }
```

Code

```
2  /* Class to create Interface which is to be used by all name Panels
3      Author: [REDACTED]
4      E-Mail: [REDACTED]
5      Date: 1/22/2019
6  */
7
8  public interface PanelInterface {
9
10     //displays Hi if true is passed else hides it
11     public void sayHi(boolean flag);
12
13 }
```

Code

```
5  /**
6  *
7  * @author [REDACTED]
8  * gitID: 01
9  */
10 public class Panel_01 extends JPanel implements PanelInterface {
11
12
13
14     Panel_01()
15     {
16
17         JLabel jlabel = new JLabel();
18         jlabel.setText(" [REDACTED]");
19         jlabel.setFont(new Font("Arial",1,20));
20         setBounds(10,10,200,60);
21         setBackground(Color.CYAN);
22         setBorder(new LineBorder(Color.GRAY));
23         setVisible(true);
24         add(jlabel);
25     }
26
27     @Override
28     public void sayHi(boolean flag)
29     {
30         JLabel jlabel=(JLabel)getComponent(0);
31         if(flag)
32         {
33             jlabel.setText("<html> [REDACTED] <br/>Hi!</html>");
34         }
35         else
36         {
37             jlabel.setText("Janice Abraham");
38         }
39     }
}
```

Code

```
1  import javax.swing.*;
2  import java.awt.*;
3
4  /** Fifth tab in the frame containing 20 panels displaying
5   * students name and "Hi"
6   *
7   * @author : [REDACTED]
8   * @version : 2.0
9   * @since : 01/19/2019
10  */
11
12
13  public class Tab_69 extends JPanel implements TabInterface{
```

```
34          panelInterface.sayHi(false);
35      }
36      catch(Exception e)
37      {
38          // If empty panel, move to the next panel
39      }
40      panelCount++;
41  }
42  thread.interrupt();
43
44      }
45  };
46  thread.start();
47
48 }
49
50 @Override
51 public void stopSayingHi() {
52     if (thread != null) {
53         startStopThread =false;
54     }
55 }
56 }
```

Code

```
69     //List of panel classes to be added
70     private String[] panelNames = {"Panel_01", "Panel_12", "Panel_14", "Panel_19", "Panel_51", "Panel_78",
71             "Panel_89", "Panel_62", "Panel_37", "Panel_66", "Panel_72", "Panel_67", "Panel_64", "Panel_75",
72             "Panel_50", "Panel_XX", "Panel_XX", "Panel_XX", "Panel_XX", "Panel_XX"};
```

6

Quiz 01

```
types.Operator):
    X mirror to the selected
    select.mirror_mirror_x"
```

```
mirror_mod = modifier_obj
# mirror object to mirror
mirror_mod.mirror_object
operation = "MIRROR_X":
mirror_mod.use_x = True
mirror_mod.use_y = False
mirror_mod.use_z = False
operation = "MIRROR_Y":
mirror_mod.use_x = False
mirror_mod.use_y = True
mirror_mod.use_z = False
operation = "MIRROR_Z":
mirror_mod.use_x = False
mirror_mod.use_y = False
mirror_mod.use_z = True
```

```
selection at the end -add
```

```
_ob.select= 1
mirror_ob.select=1
bpy.context.scene.objects.active
("Selected" + str(modifier))
mirror_ob.select = 0
bpy.context.selected_objects
data.objects[one.name].sel
```

```
print("please select exactly one object")
```

```
-- OPERATOR CLASSES -----
```

Quiz 01

- Review the code of Team X = Your Team + 1
- If (Team ID = 8) review (1);
- Grade it 0 to 100 considering readability:
 1. Code guidelines
 2. KISS
 3. DRY

You are not grading your friends. I am grading your knowledge about 1, 2, and 3. **So, do not worry for your friends.**

- **Do not forget to put your names in the paper**

