AP CS A Celebrity Lab Checklist

1. ☐ Work through Lab exercises on paper for parts 1-3
2. ☐ Write Celebrity.java (2)
3. Write the PSVM in `CelebrityRunner`
    a. ☐ Create a `CelebrityGame` instance
4. Finish CelebrityGame.java (3)
    a. ☐ Data members
        i. ☐ `Celebrity` instance `gameCelebrity`
        ii. ☐ `ArrayList<Celebrity>` instance `celebGameList`
        iii. ☐ `CelebrityFrame` instance `gameWindow`
    b. Constructor
        i. ☐ Initialize data members
        ii. ☐ Call `prepareGame` helper
    c. Methods
        i. ☐ Complete `prepareGame`
        ii. ☐ Complete game size method
        iii. Complete Celebrity related methods
            1. ☐ `sendClue`
            2. ☐ `sendAnswer`
        iv. Complete validation methods
            1. ☐ `validateCelebrity`
            2. ☐ `validateClue`
        v. ☐ Complete `addCelebrity`
        vi. Complete `play`
            1. ☐ Check the size of the `ArrayList<Celebrity>`
            2. ☐ Set the current celebrity
            3. ☐ Call the `replaceScreen` method properly
        vii. Complete `processGuess`
            1. ☐ Use the `String` methods `equalsIgnoreCase(text)` and `trim()` on the parameter
        viii. ☐ Complete sendClue()
5. Complete the `CelebrityFrame` class (GUI Appendix)

a. Constructor

```
public CelebrityFrame(CelebrityGame controller)
{
    //The first line of any subclass should ALWAYS be
    super();
    this.controller = controller;
    this.panelCards = new JPanel(new CardLayout());
    this.gamePanel = new CelebrityPanel(controller);
    this.startPanel = new StartPanel(controller);
    setupFrame();
}
```

i. ☐ Leave the `super()` call where it is
ii. ☐ Initialize the controller data member from the parameter
iii. Initialize the three panels
    1. ☐ `panelCards` with a parameter of a new `CardLayout()`
    2. ☐ `gamePanel` with a reference to the controller as a parameter
    3. ☐ `startPanel` with a reference to the controller as a parameter
    4. ☐ Call the `setupFrame` helper

b. `setupFrame` helper method

```
private void setupFrame()
{
    panelCards.add(gamePanel, "GAME");
    panelCards.add(startPanel, "START");
    this.setSize(800,800);
    this.setTitle("Celebrity Game");
    this.add(panelCards);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setResizable(false);

    replaceScreen("START");

    //Must be the last line of the configuration segment to
    //If not set as true the window will not display and the
    this.setVisible(true);

}
```

i. ☐ Add the `gamePanel` and `startPanel` to the `panelCards` member with the Strings GAME and START respectively
ii. ☐ Set the size of the frame to 800 by 800
iii. ☐ Set the title to be Celebrity Game
iv. ☐ Add the panel cards to the frame
v. ☐ Set the default close operation
vi. ☐ Set the window to NOT be resizable
vii. ☐ Call `replaceScreen` with a parameter of START
viii. ☐ Set the window to be visible!!

c. `replaceScreen` method

```
public void replaceScreen(String screen)
{
    if(screen.equals("GAME"))
    {
        //If the selected screen is the game, sends the first clue t
        gamePanel.addClue(controller.sendClue());
    }
    //Sets the chosen JPanel subclass as the active class
    ((CardLayout)panelCards.getLayout()).show(panelCards , screen);
}
```

i. ☐ Check the if the parameter equals GAME
   1. ☐ If it does, call `gamePanel.addClue` with a parameter of `controller.sendClue()`

ii. ☐ Tell the `panelCards` layout manager to show the correct screen

6. Complete the StartPanel class
   a. The setupPanel helper method

```
private void setupPanel()
{
    this.setLayout(panelLayout);
    this.add(clueLabel);
    this.add(celebrityRadio);
    this.add(literatureRadio);
    this.add(answerField);
    this.add(clueField);
    this.add(startButton);
    this.add(celebrityCountLabel);
    this.add(addCelebrityButton);

    // Adds the RadioButtons to the group so only one can be selected.
    celebrityRadio.setSelected(true);
    startButton.setEnabled(false);
    typeGroup.add(celebrityRadio);
    typeGroup.add(literatureRadio);

}
```

i. ☐ Set the layout manager

ii. Add all the GUI components
   1. ☐ clueLabel
   2. ☐ celebrityRadio
   3. ☐ literatureRadio

4. ☐ answerField
5. ☐ startButton
6. ☐ celebrityCountLabel
7. ☐ addCelebrityButton

iii. ☐ Set the default type of Celebrity
iv. ☐ Enable the startButton
v. ☐ Add both RadioButtons to the typeGroup