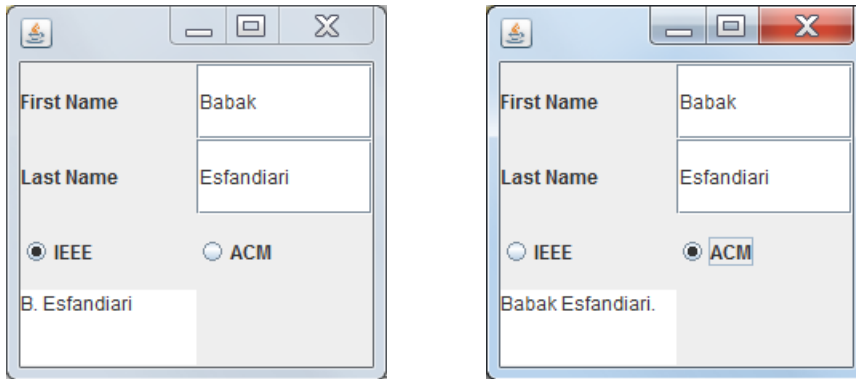


## SYSC 3110 – Lab Exam (2 hours)

We want to come up with a very simple GUI application that helps us format an article citation using different citation formats. To keep this exam brief, we are going to focus only on the formatting of the article author's first name and last name. The GUI should look like this:



The user enters the first name and the last name in the two text fields. Two radio buttons allow the user to choose the formatting strategy. If “IEEE” is chosen, the text area at the bottom shows the initial of the first name, followed by a period “.”, a space, and the last name. If “ACM” is chosen, the first name, followed by a space, followed by the last name and a period “.” is displayed.

### Hints:

- for the layout above, a simple 4 x 2 GridLayout was used (that’s why there’s an unused area to the right of the text area at the bottom);
- text fields trigger an `ActionEvent` (just like buttons!) when you hit “enter”, and so do the radio buttons when you select one. Alternatively you could add a “format!” button in the unused area to trigger the formatting. You could also decide to get fancier and update the formatting with each key press, but we won’t require it.
- radio buttons work just like regular `JButton`; you just need to group them at the end.

Here’s the official documentation along with an example:

<https://docs.oracle.com/javase/tutorial/uiswing/components/button.html#radiobutton>

### Deliverables:

- a) The complete code, packaged in a JAR file (MAKE SURE YOU INCLUDE YOUR SOURCE CODE OR YOU WILL GET ZERO);
- b) The complete UML class diagram of your design, and a UML sequence diagram showing what happens when the user hits “enter” in any field (or presses the “format!” button if that’s what you chose to do). You can use Violet to create these diagrams but make sure you export the diagram in png, gif or jpg.

Put your deliverables in a single zip file, and submit your file on cuLearn. THE DEADLINE FOR SUBMISSION IS AUTOMATICALLY ENFORCED, AND MISSING IT MEANS YOU GET ZERO. So: submit early, and submit often.

We will evaluate the following items:

- Is your program compiling and working properly? Does it crash too easily?
- Is the code well written: deals well with bad input, smell-free (no code repetition, overly long methods or complicated logic)?
- Does the design follow loose coupling and high cohesion principles?
- Are the diagrams correct and do they correspond to the code?