Pencil And Paper For Lesson 6

1. The UserIO class (see the gui folder in the supplementary code folder) uses two instance inner classes: SubmitListener and ClearListener. Here is the relevant code:

```
//code that creates instances of the inner classes
JButton submitButn = new JButton(SUBMIT);
submitButn.addActionListener(new SubmitListener());
JButton clearButn = new JButton(CLEAR);
clearButn.addActionListener(new ClearListener());

//the inner classes
class SubmitListener implements ActionListener {
   public void actionPerformed(ActionEvent evt) {
      inputString = upperText.getText();
      System.out.println("Got input: "+inputString);

   }
}
class ClearListener implements ActionListener {
   public void actionPerformed(ActionEvent evt) {
      lowerText.setText("");
      System.out.println("Clearing output text area.");
   }
}
```

   a. Explain why these inner classes should *not* be made *static nested* classes.

   b. Rewrite the code above so that the `SubmitListener` is implemented as a *local* inner class and the `ClearListener` is implemented as an *anonymous* inner class. Hint: To create a local inner class, replace the code

      `new SubmitListener()`

      with a call to another function that you create, and in the body of that function, define your listener class.

2. In your class MyClass you have a method myMethod(). You are planning to add to MyClass an inner class that will assist the functioning of myMethod. You may wish to make this inner class a member inner class, a local inner class, or an anonymous inner class,depeding on the requirements. For each of the following situations, decide which of these options would be best and explain in eachcase.

   (i) Make it a member inner class
   (ii) Make it a local inner class, defined within myMethod()
   (iii)Make it an anonymous inner class, defined within myMethod()

   A. Situation: Your method myMethod() will make use of the inner class several times, but the inner class will not be needed for any other purpose

B. Situation. Your method myMethod() will use the inner class just once, and it will not be referred to again; also, users of myMethod will make relatively few calls to myMethod()

C. Situation. Your method myMethod() will be accessed and repeatedly called in a loop from an instance of another class.