# Lab Practice Week 7

You need to show working versions of your answers to all questions to your tutor. Your tutor will expect to see them by your next session.

**What to submit:** your answers to exercises 1, 3, and 4.

***Note: even though you only need to submit those exercises mentioned above, you should attempt all exercises in each lab practice to help broaden your understanding; this may also help with your assignment work.***

**Do all the programs in NetBeans IDE.**

**NOTE: Include internal documentation in your code, if you are not sure, read chapter 2.4 in your textbook and talk to your lecturer**

---

Exercise 1 below will use the String comparison method that you did for exercise 3 of Lab Practice 2. Exercises 3 will give you opportunity to utilize the above method in a GUI application.



Before starting the exercises, make sure you have read the relevant lecture material.



1. Using your solution to question 3 of Lab Practice 2, modify it to create a class that contains a static method; the method takes a string as a parameter and returns a boolean value indicating whether the parameter string has repeated characters in it or not. That is, return `true` if there is at least one character which appears more than once in the string. The string should be input by the user.



2. The GUI program *SimplApp.java* is available from Topic 7 lecture slides. Copy the code and make a project; get the program to work. Without needing to understand everything the code does, you should at least understand the working of the constructor:

```
public SimpleApp(String windowTitle) { ... }
```

3. Copy and rename *SimplApp.java*; then modify it to create a GUI program that accepts a String from the user in a *TextField* and reports whether or not there are repeated characters in it. Thus your program is a client of the class which you created in question 1 above.

   N.B. most of the modification needs to occur in the constructor and the *actionPerformed()* methods. So you should spend time working out exactly what these two methods are doing, so that you can make the necessary modifications.

4. This exercise extends exercise 1 from Lab Practice 5. Write a method that takes an integer array as its parameter and sorts the contents of the array in ascending order using the Insertion Sort algorithm (see Topic 6 lecture notes for the algorithm). Call this method (created in exercise 1 of Lab Practice 5) after the original array and other stats have been displayed. Once the array has been sorted by your method, display its contents to the screen in the same manner as the original array was displayed.

5. The GUI program *BinarySearch.java* is available from Topic 7 lecture slides. Copy the code and make a project; get the program to work. You should attempt to understand the operations of the Binary Search algorithm.