

## Lab8: Groups of Objects - Object Behaviour

Answer each question in a different directory, named Lab8-[number]. Make sure that you include all the .java files we need in order to run your code. Unless otherwise specified the class that contains the main method should be named Lab8[number]Test We should be able to compile and run your code on LCPU (e.g. by typing `javac *.java` and `java Lab8[number]Test`).

All exercise directories should have ;username;-Lab8 as their parent directory. This folder should be compressed into a single file prior to submission. File submissions to moodle must be a zip or tar archive, the file name should be of this form:

`<username>- lab<number>.tar.gz` or `.zip`

**For example:** `ddk20-Lab8.zip` or `ddk20-Lab8.tar.gz`

There are often various ways of solving an exercise and some parts are left intentionally vague. In those cases, you can decide how you code the exercise.

If you have questions about this lab sheet or are stuck with one of the questions talk to the tutors in the lab or post your question on Moodle. If your question needs to include code, please email it to the mailing list rather than posting it to Moodle.

This lab is part of your lab coursework. **The deadline for submission is 10am Tuesday 4 December 2012**

The code associated with the this lab sheet can be downloaded from Moodle.

### 1 Adapting Existing Code

**Exercises 1.** Implement a search method in the Notebook class on the basis of a given String (parameter). Demonstrate in your tester class that your code is working correctly.

*15 marks*

**Exercises 2.** Adapt the Circle class from Lab7 by adding add a field that counts how many instances of the class have been created. Each Circle object that is being created needs to have an ID-number equal this counter at the time of creation. Demonstrate in your tester class that your code is working correctly.

*15 marks*

**Exercises 3.** Write a `toString()` method for the `Circle` class that prints out the state of a `Circle` object. Create a new class, `CircleList`, that will use a `Vector` to store `Circles`. You will also need the `Canvas` class from the `Shapes` project if you want to use the `makeVisible()` method. `CircleList` should have a constructor that creates a vector using `Vector()`. It should have a method that adds the `Circles` to the end of the vector. Another method should add a circle object to a specified index. A third method should remove a circle from a specified index. A final method should clear (delete all elements) the vector. At the end of each method print out the details (state) of each of the objects in the vector using the `toString()` method as well as their index (this should be done using a separate `printDetails()` method). Use a different iteration technique for each method. Arrays, ArrayLists and Vectors are three examples of collections objects. The `Vector` class is defined in the package, so you need to include the import statement `import java.util.Vector;` to use it. For more information on the `Vector` class take a look at the Sun Java docs. Demonstrate in your test class that your code is working correctly.

*30 marks*

## 2 Writing Classes

**Exercises 4.** Create a public class `NumberList` which declares a field (named `list`) with type array of doubles. The array should be created by a constructor which is parameterized with the length of the array. Add a method `print` which prints the array as a sequence of numbers. Add to this class a method `randomFill` which fills the array in field `list` with pseudo-random numbers of type double. Hint: there is a Java API class for pseudo-random numbers. Demonstrate in the main method of your class that your code is working correctly.

*20 marks*

**Exercises 5.** Write a static method `isPrime(int n)` that returns true if the parameter `n` is a prime number, and false if is not. A number `n` is prime is the number can not be divided by an number smaller than `n` with the exception of 1. Demonstrate in the main method of your class that your code is working correctly.

*20 marks*

## 3 Optional Exercises

**Exercises 6.** Change your notebook project so that notes are numbered starting from 1, rather than 0. Make sure you modify the `showNote` and `removeNote` methods appropriately. Demonstrate in your tester class that your code is working correctly.

**Exercises 7.** Implement a static method that breaks a `String` up into tokens separated by `“.”`, `“-”` and `“@”` (on the command line). As always demonstrate that your code works correctly using the main method. Hint: You might want to use some functionality provided by the Java API.