Australian National University

- COMP2100/6442
- Lectures
- Labs
- Activities

Lab 1

COMP 2100/6442: Lab 1 - Tools for Software Development

The objective of this lab is to get familiar with *ssh*, *gitlab* and a powerful text editor while getting a quick refresher in Java programming.

Tutorial Group Discussion [2 marks participation]

- Move away form the computers forming a semi-circle around the white board.
- Go around the group and introduce yourself.
- Ask the person next to you what they hope to get out of the course.
- What content was covered in the previous weeks lectures?
- In terms of the lecture content what was easy? What was hard?
- What are some security implications of using ssh?
- Why do you think git has become so popular?
- Software design has some similarities with design that takes place in other engineering disciplines (such as building design). What are the similarities? How is software design different to the type of design you would do in say designing a building?
- Your tutor will go over the rest of the class explaining what is expected.

The very basics [2 marks]

SSH & GITLAB

On a Linux machine using the command line:

- Set up an ssh public/private key pair that you can use for the gitlab server. Add your public key to the gitlab server at gitlab.cecs.anu.edu.au
- Create a “helloworld” project in CECS gitlab
- Clone this repository into a directory on a Linux machine

Text editor

It is crucial to be familiar with a powerful text editor or IDE to be productive while writing code. There are two kinds of text editors, those that run on the terminal such as *vim, nano, or emacs*; and those that have a GUI such as *gedit*. An IDE goes further than a text editor and provides features that allow a programmer to debug and run code from within the IDE. A very popular and common IDE for Java programming is *eclipse*.

You should be comfortable using a text editor and an IDE.

- Open a HelloWorld.java file using your favorite text editor or IDE
- Write a simple program that prints “Hello World!” to standard output. Compile and run this program
- Add and commit this HelloWorld.java file to local git repo and then push it to the gitlab server

Some more programming and resolving conflicts in GIT [2 marks]

Java Programming

- Create a project called *lab_work* on your CECS gitlab server
- Add your tutor as a master to this project
- Write a program in Java to simultaneously compute the maximum and minimum values in an array. Use the ArrayList data structure. Use a random number generator to initialize values in the array.
- Add and commit this program to the *lab_work* project
- Compile and run this program and see that it generates the correct output

GIT

Using the command line on a Linux machine:

- Create a project and clone it to two different directories. Now you have two clones of the project on your local machine.
- See how a change in one repo can be reflected into the other repo on your local system (via the gitlab server)
- Modify the same line in both of the local repos to create a conflict, resolve this conflict
- Login to the gitlab server and have a look at the projects “Files”, “commits”, “network”, and “graphs”.
- Click on the nodes in the “network” view to see the difference a particular commit has made. How can you see the conflict being resolved?

Java Programming [2 marks]

For each of the below programming problems place your solutions in a different directory. All this code must be added to your *lab_work* repo, committed and pushed.

Problem 1

Create a recursive method that calculates the *n* th Fibonacci number.
A fibonacci number is calculated by adding the previous two previous fibonacci numbers (so F(n) = F(n-1) + F(n-2)). We assume that F(0) = 0 and F(1) = 1, this

gives the two base cases for our recursive implementation.

Using this method implement a program that accepts as input the value of *n* and outputs fibonacci number F(n). example, given `6`, the program must print `8`.

How big can you make *n* before it gets very slow? Add your answer to a comment to the code.

Problem 2

Create an interface called **TextDraw** that has a method signature “void draw();”. Create two classes, one called **Box** and the other **Triangle**. Both must implement the **TextDraw** interface. The *draw()* method in **Box** must draw a box to standard out using the ‘#’ character. Whereas, **Triangle** triangle must draw a triangle. Create a class called **DemoTextDraw** which only has a *main()* method that contains:

```
   TextDraw box = new Box();
   TextDraw tri = new Triangle();
   box.draw();
   System.out.println("-------------------");
   tri.draw():
```

This would output:

```
####
####
####
------------------
#
##
###
####
```

Problem 3

Print the following quadrilateral pattern, where *n* depicts the maximum horizontal width. For example, if *n=5*, the following pattern is printed:

```
  #
 ###
#####
 ###
  #
```

The program must accept *n* as input. Assume that *n* is an odd integer with values ranging between [3,15]

Problem 4

For this problem you should be able copy your code from Problem 2 as a starting point.

Create a class called **Shape** that has an abstract method signature “void draw();”. Create two classes, one called **Box** and the other **Triangle**, both extending the **Shape** class. The *draw()* method in **Box** must draw a box to standard out using the ‘#’ character. Whereas, **Triangle** must draw a triangle. Create a class called **DemoTextDraw** which only has a *main()* method that contains:

```
   Shape box = new Box();
   Shape tri = new Triangle();
   box.draw();
   System.out.println("-------------------");
   tri.draw():
```

This would output:

```
####
####
####
------------------
#
##
###
####
```

- What is the difference between the approach taken in problem 3? Put your answer as a comment in the code.

Before you get these problem marked off, commit and push to your *lab_work* repo.

Surprise Challenge Task [2 marks]

- Once you have completed all the other tasks your tutor will allocate a surprise challenge task.

UPDATED: / RESPONSIBLE OFFICER: **Head of School**/ PAGE CONTACT: **Dr Eric McCreath**