



# MACQUARIE UNIVERSITY

*Faculty of Science*

## COMP125 Fundamentals of Computer Science Mixed Class Exercises Week 2

### Learning outcomes

By the end of this session, you will know some of Java basics. In particular, you will be able to:

- a. write, edit and run a program using Eclipse;
- b. perform simple computations, involving different types;
- c. deal with boolean statements and operators;
- d. use loops to solve simple problems.

### Questions

Most of these questions ask you to write a Java program. For each program, start a new Java project in Eclipse and create a new class to hold your code. When you create the class, check the boxes to create “public static void main(String[] args)” and “Generate comments”. Then you should write your code inside the main method as we did in the examples in the lecture.

Try running your code via the green run button and via the debugger. Remember that to get your code to stop when running in the debugger you need to insert a breakpoint. To do this, right click in the left margin of the editor window near the start of your main method and select "Toggle breakpoint". You should see a small blue blob in the margin showing where the breakpoint is set. The debugger will then stop here when you run your code via the little bug button.

1. During a storm, you always see the lightning first and then you hear the sound of thunder. This is because sound travels much slower than light. Namely, sound travels approximately at 340 m/s as compared to 300,000 km/s for light. To a first approximation we can say that the time for the flash to reach us is zero, so the time between the flash and the sound of thunder is the time that the sound takes to reach us at 340 m/s.

Based on this observation, write a Java program that calculates the distance to a storm in meters given the speed of sound and the time between the lightning flash and the sound of thunder.

2. Write a Java program to test if a date entered by the user is valid. Given two integers denoting the day and the month, your program should then be able to decide if this date is valid or not.

Your program should output different messages depending on the kind of problem with the date. Any invalid date should begin with "Invalid" followed by a reason, eg. "Invalid: February has 28 days".

3. What is the contrary of the following statements:

- She is  $> 20$  and  $< 45$  with long hair;
- He is either Spanish or Italian and he likes soccer.

4. Write a Java program that given a character ( $c$ ) and two positive integers ( $n$  and  $m$ ), draws a rectangle filled with the character  $c$  of size  $n \times m$ .

5. Write a Java program that given a character ( $c$ ) and a positive integer ( $n$ ) draws a pyramid filled with the character  $c$  of height  $n$ . First, design your approach with pen and paper only. Then implement your approach.

Modify your program to draw a diamond instead of a pyramid.

#### 6. Programming Exercise to be Submitted

*Your solution addressing the following problem is to be submitted as a Java program via iLearn (Moodle) by Saturday Aug 13, 2011 noon.*

*Your submission will be graded and is worth 1 mark towards your final grade for COMP125. Grading will occur in your mixed class - show your solution to your tutor when you are ready. You may demonstrate your solution either this week (week 2) or next week (week 3).*

Write a Java program to test whether a given string is a palindrome. A palindrome is a string that reads the same forwards and backwards, for example “lol”, “poop” and ”wow” are all palindromes.

You may solve this problem by writing code entirely in the `main` method in your Java program, as in most of the examples we did in the lecture. Define a string `mystring` and then test whether it is a palindrome, printing out a message at the end.

If possible though, you should define a static method that takes a string argument and returns a boolean value of `true` when that string is a palindrome and `false` otherwise. In the same way that I took the code for the vowel counter in the lecture and turned it into a static method (procedure), you can do the same with your solution. The static method would look like this:

```
1     private static boolean is_palindrome(String mystring) {
2         // TODO Auto-generated method stub
3         return false;
4     }
```

Either solution is ok, but you are encouraged to write your code as a static method.

When you have your code working, test it using different input strings - some right, some wrong. In particular make sure you get the right answer for: lol, poop, wow, foaf, can and thisisnotapalindrome.

According to your program, is “a” a palindrome? Do you think it’s correct?

**Challenge:** if that’s all too easy. Write a second static method that tests whether a string is a palindrome but ignores whitespace, punctuation and capitalisation in the string. Then you can test longer palindromes like “A man, a plan, a canal, Panama!”.