LAB 3

Suggested order: Complete part 1 before completing part 2,3,4

Part 1

After completing this week's practical, you should be able to use the if statement to get your program to respond differently in different situations. See lecture slides for details on using if statements in your programs.

In this part you should write a program which gets two integer (`int`) numbers from the user using two `JOptionPane InputDialog` pop-up windows, and compares the two numbers. If the **first** number is less than the **second**, the program should use a `MessageDialog` to display the message

The first number (?) is less than the second number (?)

Where the first question mark represents the first number entered and the second question mark represents the second number. If the **second** number is less than the **first**, the program should use a `MessageDialog` to display the message

The second number (?) is less than the first number (?)

Where the first question mark represents the **second** number entered and the second question mark represents the **first** number.

Finally, if the two numbers are equal, the program should use a `MessageDialog` to display the message

The two numbers you entered are equal

As before, your program should read each number entered in the `InputDialog` into a String variable (everything entered in an `InputDialog` is a String, and so must go in a String variable), and should then convert that String into an `int` and put it into an `int` variable. See lecture notes on Fundamental Data Types for details on converting from a String to an int.

Remember, to use `JOptionPanes` in your program, there must be a line at the start of your program Importing the `JOptionPane` class (see notes on Communication with JOptionPanes for details on `JOPtionPanes`). Finally, finish off your program with the System.exit(0) statement (to ensure that your pop-up windows close correctly and the program ends properly).

Your program should consist of a single class with a single main part. You should call your class `CompareTwoNumbers`. You should save your program in a file with the same name as the program's class name. When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 1 of that practical for details on compiling and running programs). Make sure your program compiles and runs correctly; if any errors are produced by your program, fix them, save the file again, and recompile.

Part 2

In this part you should write a program which gets two integer (int) numbers from the user using two JOptionPane InputDialog pop up windows. The program should first put the numbers into two int variables (converting the typed in String to an int as in notes on Fundamental data types). The program should then compare these two variables. If the number in the first variable is less than the number in the second variable the program should:

. put the first number into an int variable called lower

. put the second number into an int variable called higher

Otherwise, if the first number is not less than the second number, the program should

. put the second number into an int variable called lower

. put the first number into an int variable called higher

Once the program has correctly assigned the numbers to the variables higher and lower, the program should find out if the lower number is a factor of the higher number (that is, if the lower number divides evenly into the higher number, with no remainder). If the lower number is a factor of the higher number, your program should work out how many times the lower number goes into the higher, and display the MessageDialog

? is a factor of ? (? goes into ? exactly ? times).

In this message, the first question mark is the lower number, the second is the higher number, the third is the lower number again, the fourth is the higher number again, and the final question mark is the number of times the lower number divides into the higher number. For example, if the user entered the numbers 221 and 17 (or 17 and 221), the message displayed would be

17 is a factor of 221 (17 goes into 221 exactly 13 times).

(17 * 13 = 221). If the lower number is not a factor of the higher number, your program should work out how many times the lower number goes into the higher, and what the remainder is. Your program should then display the MessageDialog

? is not a factor of ? (? goes into ? a total of ? times, with ? over).

In this message, the first question mark is the lower number, the second is the higher number, the third is the lower number again, the fourth is the higher number again, the fifth question mark is the number of times the lower number divides into the higher number, and the final question mark is the remainder. For example, if the user entered the numbers 218 and 17 (or 17 and 218), the message displayed would be 17 is not a factor of 218 (17 goes into 218 a total of 12 times, with 14 over).

(17 * 12 + 14 = 218). In deciding whether one number goes evenly into another, and if there is a remainder figuring out what it is, you should use %, the modulus operator. a % b will compute the remainder left after a is divided by b (if b is a factor of a; that is, goes evenly into a, this remainder will be zero; in other words, if ((a % b) == 0), it means that b divides evenly into a, with no remainder).

Your program will need two if else statements, one after reading in the numbers (to decide which number should go into the lower and which into the higher variable), and another starting

after the end of the first if□]else statements (to decide whether the lower number is a factor of the higher or not).

Remember, to use JOptionPanes in your program, there must be a line at the start of your program Importing the JOptionPane class. Finally, finish off your program with the System.exit(0) statement (to ensure that your pop□]up windows close correctly and the program ends properly).

Your program should consist of a single class with a single main part. You should name your class Factors and save your program in a file with the same name as the program's class name.

When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 1 of that practical for details on compiling and running programs). Make sure your program compiles and runs correctly; if any errors are produced by your program, fix them, save the file again, and recompile.

Part 3

In this practical you will learn more about using "nested" if statements (if statements that have other if statements inside them).

In this practical you should write a program that can compute the area of a circle, a rectangle, or a triangle. Your program should start by using a JOptionPane InputDialog to ask the user whether they want to work out the area of a circle, rectangle or triangle. The user should enter the word circle if they want to get the area of a circle, the word rectangle if they want to get the area of a rectangle, and the word triangle if they want to get the area of a triangle.

If the String entered by the user is equal to the String "circle", the program should use a JOptionPane InputDialog to ask the user to enter a radius value. The program should convert the string entered by the user into a double value, and store that in a radius variable. If the radius value is less that 0, the program should use a JOptionPane MessageDialog to display the message

Radius value less than zero: cannot compute circle area.

If the radius is not less than zero, the program should work out the area of a circle with that radius, and should use a JOptionPane MessageDialog to display the message

A circle of radius ? has an area of ?

where the first question mark represents the entered radius value, and the second represents the computed circle area.

If the String entered by the user is equal to the String "rectangle", the program should use two JOptionPane InputDialogs to ask the user to enter the two sides of the rectangle. The program should convert the String entered by the user into double values, and store them in two variables. If the one or other of these values is less that 0, the program should use a JOptionPane MessageDialog to display the message

Side less than zero: cannot compute rectangle area.

If neither of the sides are less than zero, the program should work out the area of a rectangle with those sides by simply multiplying the two side values, and should use a JOptionPane MessageDialog to display the message

A rectangle of sides ? and ? has an area of ?

where the first question mark represents one entered side value, the second represents the other side value, and the third represents the computed rectangle area.

If the String entered by the user is equal to the String "triangle", the program should use two JOptionPane InputDialogs to ask the user to enter the width of the base and the perpendicular height of the triangle. (The area of a triangle is equal to half the base length by the perpendicular height). You should check for a 0 length as above. You should output the area of the triangle in a similar fashion to that of the rectangle.

Finally, if the String entered by the user is not equal to either circle, rectangle or triangle the program should display the JOptionPane MessageDialog

I don't recognise the shape ?: cannot work out the area.

where the question mark represents the String typed in by the user

Remember to finish off your program with the System.exit(0); statement (to ensure that your popup windows close correctly and the program ends properly). When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 2 for details on compiling and running programs). Make sure your program compiles and runs correctly; if any errors are produced by your program, fix them, save the file again, and recompile. You should call your class CircleRectangleTriangle and save in a file with the same name. Make sure you test every possible choice in your program, and that your program responds correctly in each case - that it can work out the area of a circle, and responds correctly if the radius is less than zero; that it can work out the area of a rectangle, and responds correctly if either side lengths are less than zero; that it can work out the area of a triangle, and responds correctly if either the perpendicular height or the width of the base is less than zero; and that it responds correctly if neither circle, rectangle or triangle are asked for. You should try running your program with every different possible input from the dialog boxes, the make sure it responds correctly for each one.

Part 4

In this practical you should write a program which asks the user for their first name and recognises when the typed-in name is the same as the programmer's name (your name). Your program should start by storing your own first name in a String variable called programmerName. For example, if I was writing the program, the first line would be

String programmerName = "Fintan";

(you would use your own name instead of "Fintan"). After storing this special name in the programmerName variable, your program should use a JOptionPane InputDialog box to ask the user for a name. If the entered name is not the same as the name stored in programmerName, the program should use a JOptionPane MessageDialog to give the following message to the user:

Hello ?, nice to meet you.

where the question mark represents the name just typed in by the user.
If the name entered by the user is the same as the name stored in programmerName, the program should respond in a different way. First, the program should use a JOptionPane InputDialog to give the following message to the user:

Hi ? ! Tell me a secret.

where the question mark represents the name just typed in by the user. The InputDialog will allow the user to enter a "secret": this should be stored in a String variable.

Once a secret has been entered, the program should respond in two different ways, depending on how long the secret is (you can use the length method of the string class to find the length of a string). If the secret is less than 20 characters long, the program should respond:

That's not much of a secret.

If the secret is more than 20 characters long, however, the program should respond

That's a good secret! I'll never tell.

Two tips:
• Before starting to write this program, it would be a good idea to write out, on a piece of paper, a rough layout of the program, showing the different if and else parts of the program, and what is inside each one. Otherwise the ifs and elses required for the program may get complicated.
• The .equals() method of the String class pays attention to capitalisation when comparing two Strings: it will say that "Fintan" and "fintan" are not equal, for example. If you want to have your program ignore capitalisation when comparing strings, you can use the

.equalsIgnoreCase() method, which works just like .equals() but will say that "Fintan" and "fintan" are equal.

Make sure you test every possible choice in your program, and that your program responds correctly in each case. For example, for my program, I tested it with the name "fintan" and with a different name, to make sure it only asked for a secret when I gave it my name. When the program asked for a secret, I tested it with a short secret ("I'm 33 years old"), and with a long secret ("I'm not a very good programmer"), to make sure it responded correctly in each case. You should test your program in a similar way.

Remember to finish off your program with the System.exit(0); statement (to ensure that your pop-up windows close correctly and the program ends properly).

Call your class MySecret and save it in a file with the same name. When you have written and saved your program, you should compile it and run it using the command prompt as in practical 1 (look at part 2 of that practical for details on compiling and running programs). Make sure your program compiles and runs correctly; if any errors are produced by your program, fix them, save the file again, and recompile.