Lab 3 (20 points total)
The purpose of this lab is to continue learning and reinforcing Chapter 3 concepts. There are several learning objectives to this assignment

- Various if statements including if else, if else if, and nested if
- Switch statements
- Data formatting with printf and/or String.format
- Incorporating documentation and style into your code

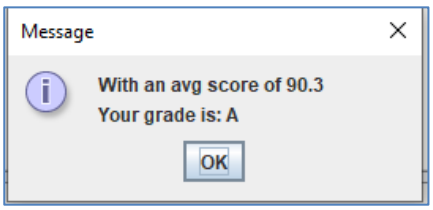
Prog Chall 1 - pg 181 (2pts) – You must use a **switch** statement to get full credit for this problem. **Reduce the valid number range to be between 1 and 3** (vs 1 and 10 in the book). Include a (default) condition for all integers outside of 1 ->3 and make sure to use break; for each case to avoid 'bleed' through all the remaining cases. **HINT – Check out section 3.9 in the book, pg150+ with a focus on pg 153**

At a minimum, use the following test cases: (1) 1 (2) 3 (3) 4

Prog Chall 4 – pg 181 (4pts) – You must use a GUI for each of the three exam score inputs JOption.showInputDialog(). Remember JOptionPane reads in **Strings** so you will need to convert your inputs to numbers in order to calculate the average. Your output should display the average out to one decimal place (ie 92.6) using String.format() since a JOptionPane ***along with the grade. NOTE-When dividing integers, the denominator needs to be a decimal number (i.e – 3.0) or you must use casting to convert the numerator to a double. For example, (n1+n2+n3)/3.0 or (double)(n1+n2+n3)/3. The 2nd version is better since the number of items is whole (3 grades). Version 1 works, but is not as clean.***

At a minimum, use the following test cases:

(1) 90, 95, 86 – should produce **`Avg = 90.3, Letter Grade = A`**
(2) 89, 70, 80 – should produce **`Avg = 79.7, Letter Grade = C`**
(3) 76, 50, 52 – should produce **`Avg = 59.3, Letter Grade = F`**



Sorting – 4pts
Write a program to read in three non-negative integers from the keyboard. Display the numbers in increasing order. You can assume that the same number will not be entered twice. ***You must use Nested ifs to solve this problem.***

At a minimum, use the following test cases: (1) 3, 2, 1 (2) 2, 3, 1 (3) 1, 3, 2.

ValidDate – 10pts
Write a program that reads a String from the keyboard and tests whether it contains a valid date. Display the date and a message that indicates whether the date is valid or not valid. **If the date is not valid, also display a message why the date is not valid** (ie, "2/29/1900 is not valid. cannot have 29 days in a non-leap year").

The input date will have the format *mm/dd/yyyy*. *Hint use Scanner method .useDelimiter("/") to change the default delimiter from a space " " to "/". useDelimiter() is not a return type method (we will learn more about methods in a few weeks), and does not have to be assigned to a var. So, if a Scanner object named keyboard was defined,* `keyboard.useDelimiter("/");` //changes the default delimiter to "/".

A valid month value *mm* must be from 1 to 12 (January is 1). The day value must be from 1 to a value that is appropriate for the given month. April, June, September, and November each have 30 days. February has 28 days except for leap years when it is 29 days. The remaining months have 31 days. An easy way to know 31 day months is to count knuckles and in between knuckles on your hand. Knuckles are 31 days and in between are non 31 day months. Also, a leap year is any year that is divisible by 4 && NOT divisible 100 (2004 is a leap year, 1900 is not a leap year) || divisible by 400 (2000 is a leap year). NOTE: This is a great review problem of

major concepts covered to date. The output should print out the date evaluated and either indicate it is valid or if not valid why. See examples below.

Test the following sets of inputs:

1) 2/29/2000
2) 2/29/1900
3) 2/29/2020
4) 4/31/2021
5) 4/15/2021
6) 7/25/2021
7) 5/32/2013
8) 13/13/2013

Provide output:

1) `2/29/2000 is a valid date.`
2) `2/29/1900 is not a valid date. Cannot have 29 or more days in a non-leap year.`
3) `2/29/2020 is a valid date.`
4) `4/31/2021 is not a valid date. Cannot have 31 days in this month.`
5) `4/15/2021 is a valid date.`
6) `7/25/2021 is a valid date.`
7) `5/32/2013 is not a valid date. Cannot have more than 31 days in any month.`
8) `13/13/2013 is not a valid date. Months must be 1 to 12.`

Submitting your work

For all labs you will need to provide a copy of all .java files. **No need to provide .class files. I cannot read these.** ***NOTE – For Replit, please update Main.java to another name such as ProChal1.java, ProChall3.java, etc.*** In addition to your .java files, you will need to provide output files of your console. The name of the output file should match the class name and have the .txt extension such as ProChal1Out.txt, ProChall3Output.txt. For GUIs such as JOptionPane, you will instead need to create screenshots. For Windows users, Snipping Tool is a great way to do this. Chromebook - Shift+Ctrl+Show Windows. Mac OS users, you can see how to take screenshots using the following url - https://support.apple.com/en-us/HT201361.