COMP108 Data Structures and Algorithms
Lab Exercises (Week 2)
Due: 19 February 2021, 5:00pm

(Late submission not allowed)

Information

- Submission: **Submit the file COMP108W02.java to SAM**
  https://sam.csc.liv.ac.uk/COMP/CW_Submissions.pl?qryAssignment=COMP108-12
  Late submission is not allowed.
- Submission of lab/tutorial exercises contributes to 10% of the overall module mark. Submission is marked on a pass/fail basis - you will get full marks for submitting a *reasonable attempt*.
- Late submission is **NOT** possible. Individual feedback will not be given, but solutions will be posted promptly after the deadline has passed.
- These exercises aim to give you practices on the materials taught during lectures and provide guidance towards assignments.
- Relevant lectures: Lectures 2 & 3

1. **Programming — Preparation**

   You have been asked to prepare your programming environment last week. If you haven't done that yet, follow the discussion on Canvas:

   https://liverpool.instructure.com/courses/19756/pages/compiling-and-running-java-programs

   You can use web IDE: https://ide.cs50.io/ if you haven't setup your own environment.

   (a) Download two java files "COMP108W02App.java" and "COMP108W02.java" from Canvas via the link "Labs & Tutorials".

   (b) Open the files with a text editor (e.g., notepad++ or web IDE editor).
   Beware of where you have saved the java file and open it from the correct folder. Do NOT use MS Word!

   (c) COMP108W02App.java contains the class COMP108W02App and takes care of data input. The algorithms to be implemented are in the file COMP108W02.java which contains the class COMP108W02.

   (d) Open a command prompt (cmd) and change to the folder where you saved the programs.

   (e) Compile the programs by typing first **javac COMP108W02.java** and then **javac COMP108W02App.java**. There should be two files created: COMP108W02.class and COMP108W02App.class.

   (f) Run the program by typing **java COMP108W02App**.

   (g) Test the program by entering two positive integers (with the second integer larger than the first one). For simplicity, the program does not validate input.

   (h) You are going to work on COMP108W02.java in which some methods are already implemented: `sumFromOne()` and `isFactor()`. You are going to complete three methods: `sumFromTo()` for Task 1, `multipleFactor()` for Task 2, `bugOne()` for Task 3.

2. **Programming — Task 1**

   (a) Insert your name and student ID at the beginning of COMP108W02.java.

   (b) Study the method `sumFromOne()` that has been completed. The method takes two parameters `number` and computes the sum of integers 1, 2, 3, $\cdots$, `number`.

   (c) Edit the method `sumFromTo()` to compute sum of integers from `numberX`, `numberX+1`, $\cdots$, `numberY`.

   (d) Compile, run, and test your program to see if it computes the sum correctly.

   Test cases to try:

X	Y	$\sum_{i=1}^{X} i$	$\sum_{i=1}^{Y} i$	$\sum_{i=X}^{Y} i$
3	5	6	15	12
2	10	3	55	54
9	10	45	55	19

3. **Programming — Task 2**

   (a) Continue to work on COMP108W02.java. The method `isFactor()` has been implemented.

   (b) Add to the method `multipleFactor()` to find all multiples of `numberX` that are factors of `numberY`.

   (c) Test cases to try:

X	Y	output
10	30	10 30
6	30	6 30
10	100	10 20 50 100

4. **Optional Programming Task — Fixing bugs** [do this if you finish the other programming tasks]

   (a) Continue to work on COMP108W02.java in particular the method `bugOne()` which attempts to output all common multiples of two entered numbers x and y up to 100. The method contains bugs and can be fixed by altering **ONE** line of code. Try to fix it if possible by altering only one line of code.