**Introduction To Programming**
**Tutorial 4**
(See Canvas→Assignments for due dates and marks)

**Note:** Do not include your name, student ID or any personally identifiable info in your submission as the submission may be used for peer reviews; your submission will not be lost as Canvas keeps track of these internally.

Please follow all of the steps below in the given sequence:

1. Read all unread announcements and unread replies to announcements under Canvas→Announcements.

2.1 Do any missed tutorials before going further.

2.2 Watch any unwatched recordings of the compulsory **Weekly Live Lecture** and any important videos in the Extra Videos Playlist.

2.3 If you need help in addition to what has been shown in the compulsory weekly live lecture, you are also expected to speak to your **group tutor via discussion forums** and attend/watch their live sessions. Please note that group tutors cannot debug your assessment code on your behalf as debugging is a part of every programming assessment.

2.4 **If you still have have any unresolved questions** or **if you need further feedback**, post the relevant parts of your submitted work in a new post under Canvas→Discussions→Tutorial discussions and ask from your group tutor. E.g. you can ask "*In the live lesson Gayan did ___ with ___. I didn't do ___ so should I be doing this as well?*", etc. Please note that the university requires teaching to be conducted in an equitable manner so your tutors will require you to post questions in the discussion forums.

3. Check any available feedback of your previous submissions and if you have any unresolved questions or if you need further feedback, post the relevant parts of your submitted work in a new post under Canvas→Discussions→Tutorial discussions and ask from your tutor. E.g. you can ask "*Gayan showed _____ but I did mine like _____, so which is the better approach and why?*", etc. Please note that the university requires teaching to be conducted in an equitable manner so please only use email for matters such as special consideration.

4. Follow the materials under Canvas→Modules→Week 4…

5.1 Extend the `CPT120GradeMaker` (refer to solution from last week, if necessary) so that the user enters how many marks they would like to process. Your program should then use a while-loop in the format of 'scenario 1' under Modules→Week 4 lesson PDF to achieve this functionality. Explain in comments what your code would do if the number of marks entered is negative (you can assume only integers will be entered by the user).

5.2 After completing 5.1 above, make a copy of your `main` method and rename one of the copies to `main51` (so that it does not run as the default main method). Please refer to Gayan's Monday lectures if you are not familiar with this approach.

The copy that was not renamed, which we will use for 5.2, should remain as `main`, so that it will run when the program is run. Now change the code in the main method to repeat indefinitely (see 'scenario 2' under Modules→Week 4 lesson PDF). The rationale is not require the user to enter how many marks they would like to process beforehand.

This approach requires you to decide "when to keep repeating". E.g. after a user enters a particular student's mark, you can ask the user if they wish to process another student and then repeat the code. Alternatively, you can tell the user to select the 'cancel' button (on JOptionPane) or enter an blank string (suitable for Scanner, see String class documentation) or either to finish the processing.

In CPT120 Introduction To Programming, you must only use while-loops unless otherwise required by a question. Your while-loop's condition must eventually fail. The condition must be descriptive of when the loop will continue (e.g. saying `while(keepGoing==true)` is not an example of a descriptive condition; have the continuation criteria in the condtion as shown in 'scenario 2' of the week 4 modules lesson PDF. You must not use `break`, `continue`, `return`, `System.exit`, etc. or similar branching in the middle of methods, loops, if-statements, etc. as doing so is referred to as spaghetti code.

**Optional:** Validate the inputs so that negative values are rejected (e.g. see "age" validation example in week 4 modules lesson PDF). You may assume that the user will only enter values in the requested data type (e.g. will not enter text when a int is asked for, etc.).

More steps on next page...

**Submission Checklist:**
1. Ensure that your code does not have any red dots (Java errors) as code with such errors cannot be tested/marked and will receive 0 for that submission. If your code has red-dots, refer back to similar code and fix the error or remove the code that is causing the problem. You must not leave any commenteded out code in your submissions. Yellow dots are warnings and these are different.
2. Ensure that you have added comments to your .java file explaining what you have done and any potential alternative approaches.
3. Format your code (e.g. Eclipse→Source→Format).
4. Go to Canvas→Assignments→**Independent Investigative Effort 4** and select 'submit assignment'.
5. Select to attach files from your computer, navigate to your Eclipse workspace folder→Project folder→src folder and select the (one) final version of your `CPT120GradeMaker.java` file. Your file should contain a `main` method (corresponding with step/exercise 5.2 and another named `main51` (corresponding with exercise 5.2). Please **do not submit more than 1 file** as it delays the marking process. You can also context select on the .java file name from package/project explorer inside Eclipse and find its exact location. Only the last submission is the official submission.
5. Verify your submitted files as shown during the week 1 session.

Having trouble with usernames, passwords, access, etc.? Please call the RMIT IT Service and Support Centre for quick help on 03-9925 8888 and remember to ask for a reference number and pass it on to your instructor.

Need extensions or special consideration? Please follow details and process below:
https://www.rmit.edu.au/students/student-essentials/assessment-and-exams/assessment/special-consideration