Introduction To Programming
Tutorial 5

(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 5…

For these exercises, you are expected to show your work that is in progress and ask questions from your group tutors via Canvas→Discussion forums as needed. Please note that tutors are not able to give answers before the deadline or conduct one-on-one tutoring via email or direct messaging as these are not equitable practices. When posting code, please replicate any errors or issues in isolation rather than putting the whole code up there and asking others to fix it; doing the former will also help you abstract the problem as well.

5.1. Implement code in the Modules→Week 5→Arrays lesson PDF named '*full program demonstrating the basics*' in a new Eclipse project, rename it as SalaryProcessor and get it to work. Now modify it so that the array size (originally 3) is taken as a user input. You must use a *while* loop to ensure that only integers that can be used for creating an array are accepted (see while loop lesson on indefinite looping). Afterwards, using a while loop, take inputs and assign values to each array position.
Optional: You can make relevant changes to use JOptionPane instead of System.out…

5.2. Modify the code from exercise 5.1 so that job names are also stored in an array of a data type suitable for storing names. E.g. job names array element 0 would correspond with the salary array element 0, then 1 with 1, etc. What is a good name for this array? Why? When displaying a salary at any point, display also the job name that corresponds with that salary.

5.3. [This is an investigative learning exercise. If you are unable to complete any part, you must detail the research that you did to demonstrate effort] Modify the code from the previous step to calculate the total and the average of all salaries. Finally, for each salary, display by how much it is different from the average salary. Note: Even if you are unable to complete this, you must follow your instructor's demonstration when shown during the weekly compulsory chat lesson.

5.4. [Optional to submit but must follow when solution shown] A sorting algorithm re-arranges values in an array in either the ascending or the descending order. Select the "Play All" button on the Sorting Algorithms Animations page to see how different sorting algorithms (columns) sort data in various starting orders (rows). Read the description of a sorting algorithm and implement it in your application to sort the salaries in either ascending or descending order.
Note: You will not be quizzed on sorting algorithms but they are a good demonstration of how arrays can be used.

5.5. Add comments in the style required by Assignment 2. See rubric in section 9 of the Assignment 2 PDF.

5.6. [Optional to submit but must follow when solution shown] How can we resize an array (without using ArrayLists)?

6. If you have not submitted your final version of A2, add comments explaining your plan. Note that this will not be marked but it is to help you progress.

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 5** 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 `SalaryProcessor.java` file. 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