Introduction To Programming: Independent Investigative Effort 12

Due: See Canvas→Assignments for due dates, marks and submission link.
Obtaining help method 1: Create a post in the corresponding Canvas→Discussions→IIE forum, post a screenshot of your preliminary/faulty along with a brief explanation and your tutor will help in a general way. Debugging will still need to be done by the student as it is an important aspect of programming.
Obtaining help method 2: Weekly tutor chats (not to be confused with 'live lecture') of this week will give directions (not solutions) on how to approach this IIE. Check recordings if you cannot attend.
Getting feedback: For specific questions asked, you will receive feedback via the forum within 1-2 working days by your group tutor. If you ask during a tutor chat, you would receive feedback immediately.
Late submissions: Accepted for up to 1 week with an automatic 10% penalty for each day late (unless special consideration received).
Solution: Solution, recommended approach, common mistakes to avoid, etc. for main tasks will be shown during the weekly live lecture that starts at the time of submission.
Marks: IIEs are not tests. Marks are given based on effort and not on correctness. Marks turnaround time is approx. 10 working days after submissions close.
Access issues: For non-programming technical issues (relating to infrastructure, passwords, 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.
Extensions: For all new extensions, apply for special consideration online. Contacting your tutors, instructors first will lead to delays.

Please follow/complete all steps below in the given sequence:

1. Check your official @student.rmit.edu.au email account for announcements and other communication from the university. If getting in touch with your instructors, please only use this account (not Canvas inbox, messages, personal email, phone, Microsoft Teams, etc.)

2. Watch any unwatched recordings of the **Weekly Live Lecture** and complete all missed tutorials **before going further**.
For your convenience, the time stamps of recordings are sent via student email/Canvas→Announcements.

3. Is there something that you have not fully grasped from what has been covered so far? Please have your doubts clarified via one of the relevant forums under Canvas→Discussions. Leaving gaps has shown to be severely detrimental to learning.

4. Did you want to make any additions to the previous IIE? Please do by replying to your original post. i.e. do not edit, change the images of existing posts as it affects submission timing.

5. **Coding exercise steps (Hint: Need help? Ask your tutor via Canvas→Discussions→"IIE12"):**

Complete the 15/Feb/2021 solution first as task 'a' extends upon that work. Next follow Canvas→Modules→Week 12 where the topic of comparing languages is covered (relevant to task 'b'). **Your tutors will provide further explanations on these general topics and how they relate to this IIE during their weekly "tutor chats"**. Please also follow the announcement 'How to debug large programs and get help on debugging...'

a) Create a project named 'IIE12' or similar which follows the Assignment 3 structure (i.e. Application, BackEnd, FrontEnd…, etc.) so that, if your other application were the student manager (the application used by the university staff), this new one allows a student to login and edit their own details. Both back ends (projects) should read from the same "master" database (this violates A3 requirements but it is permitted for this task).

b) Create a discussion forum post **(do not attach files)** with approximately the same algorithm, first implemented in Python and then in Java. (Your Python program does not need to run or be syntactically valid). Both programs should contain at least an if-statement and a while-loop. Alternatively do or optionally add another, the same algorithm implemented in C and Java.

Tip: Find an existing Python program online by searching in Google using the *filetype:py* operator in your Google search query then translate it to Java by yourself. Remember to give a full reference using the IEEE referencing style. Ask your tutor questions if you have any. If your tutor asks questions, you must reply to them.

Submission Checklist for Step 5:
a. Ensure steps above have been followed in sequence.
b. Ensure that there are no red dots (compilation errors) in your code. Code with red dots are not valid Java and cannot be marked.
c. If you have not made a final submission for your Assignment 3, make a dummy submission for Assignment 3 by submitting **your .java file to Canvas→Assignments→Assignment 3.** Remember, you can overwrite this submission any time when you have a proper submission for your assignment.
d. Take screenshots of the code and the data files and embed the **screenshots in a post under Canvas→Discussions→Independent Investigative Exercise 12**. If you are unable to embed screenshots, please follow the announcement 'Having issues embedding images? Here's the alternative...' The mark for this week's work will be given based on this submission.
e. Download your own file(s) from the discussion forum and ensure that it is correct. If it is not, you can edit/delete your post and retry.