Introduction To Programming
Tutorial 12

(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 12…

5.1 With the help of your group tutor via the forums, convert the Java program below to either a C or a Python (version 3) program. Your code does not need to be tested and it is fine if it has syntax errors.

```
public class MysteryProgram {
        public static void main(String[] args){
                int a=1680;
                int b=a/2;
                int c=1;
                while(c<=(a/c)){
                        if (a%c==0){
                                System.out.print(c+" x "+(a/c)+" = "+a+"\n");
                        }
                        c+=1;
                }
        }
}
```

5.2 Add comments to your C or Python (version 3) program explaining what the above program does.

Submission Checklist:
1. You must not leave any commented out code in your submissions.
2. Ensure that you have added comments to your 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 12** and select ‘submit assignment’.
5. Select to attach files from your computer, navigate to your folder and select the (one) final version of your **`MysteryProgram.py`** **`or`** **`MysteryProgram.c`** file. Please **do not submit any other files** as doing so delays the marking process.
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