Introduction To Programming
Tutorial 1 (Independent Investigative Effort 1)

(See Canvas→Assignments for due dates and marks)

Please follow all 9 steps below in the given sequence:

1. Read all unread announcements under Canvas→Announcements.

2. Live (chat) lessons

2.1: Bookmark the 'weekly lecture' link from Gayan's opening announcement:
https://rmit.instructure.com/courses/56609/discussion_topics/506159

The above link lets you join a live lesson or see when the next live lesson is taking place.

2.2: Check the 26/Aug/2019 first live lecture recording via Collaborate Ultra.

Bookmark the alternative lecture recordings playlist:
https://www.youtube.com/playlist?list=PL-GM8dbPIbtmJJkpQcZdkgbfTjVrE3Wn2

2.3: See an upcoming chat schedule (and unedited recordings) via Canvas→Collaborate Ultra

3. Communication:
For the purposes of Intro To Programming, assume that you are working in a team of programmers in a commercial environment and that your instructor is your client/supervisor. Note, however, all submitted work must be your own.

As the university requires instructors to follow an equitable mode of teaching please consider asking questions in a general form in the relevant Canvas→Discussions forums. Use email only for matters relating to special consideration. Emailed questions relating to subject matter maybe posted in the discussion forums.

3.1: Bookmark and subscribe to the discussion forums under Canvas→Discussions.
3.2: Read the guidelines for the 'Java/Eclipse General', 'Tutorial discussions' and 'Assignment 1 Clarifications' forums.
3.3: Create a test post under Canvas→Discussions→Meet your classmates.

4. Programming task:

4.1: Watch the recordings of the 26/Aug/2019 live lesson (see 2.2 above).
4.2: Follow the content and instructions under Canvas→Modules→Week 1.
4.3: Create a new Eclipse project (e.g named *Week01-Tutorial)* then create a new Java class named Myself. Copy and paste the code below and get it to run. (Hint: Watch recording of the 26/Aug/2019 lecture)

```
import javax.swing.JOptionPane;
public class Myself {
            public static void main(String[] args) {
                        String message;
                        message = "";
                        message = message + "Hello world!\n";
                        message = message + "My hobbies are...\n";
                        message = message + "Programming is used/can be used in ... for...\n";
                        JOptionPane.showMessageDialog(null, message);
            }
}
```

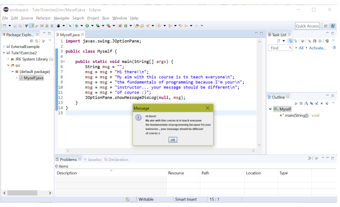
Note: It is very important that you write things exactly as given. E.g. "myself" is not the same as "Myself"; do not put spaces where spaces are not found in the examples.

4.4: Now modify program from the previous exercise in this tutorial and personalise it by stating what your hobbies or favourite pastimes and if you have seen or have any ideas on how to use programming in your hobby/pastime. Answers are not checked for correctness :)

4.5: Add code comments to your program and identify and explain the following terms in your own words: **Class**, **method, variable** and a code **block**. Listen to the 26/8 live lecture as well.

4.6: Submit your `Myself.java` and a full screenshot of your entire Eclipse window via Canvas→Assignments→Independent Investigative Effort 1 then verify your submission.

Important note on late submissions: As the tutorial solutions are discussed immediately after the deadline, a full mark cannot be awarded for submissions made after the deadline. A university standard 10% late penalty applies for 24 hour blocks from the deadline for up to 5 working days late; submissions will not be accepted this period without special consideration) Submissions made several days later may require longer time to mark as they have to be marked separately; contact your instructor if you have any issues or if you have received special consideration (see step 9 below also.)

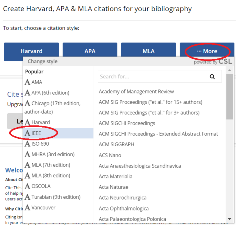


```
import javax.swing.JOptionPane;

public class Myself {

    public static void main(String[] args) {
        String msg = "";
        msg = msg + "Hi there!\n";
        msg = msg + "My aim with this course is to teach everyone\n";
        msg = msg + "the fundamentals of programming because I'm your\n";
        msg = msg + "instructor... your message should be different\n";
        msg = msg + "of course :)";
        JOptionPane.showMessageDialog(null, msg);
    }
}
```

Hi there!
My aim with this course is to teach everyone
the fundamentals of programming because I'm your
instructor... your message should be different
of course :)

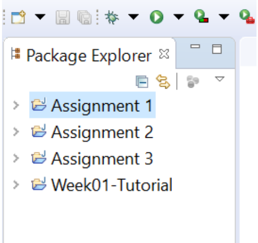
5. Referencing and academic integrity: Listen to Gayan’s explanation during the 26/08 live lecture on:

- 5.1: Why we give references.
- 5.2: When we need to references in programming.
- 5.3: How to use a reference generator tool such as citethisforme. (Tip: Choose more→IEEE)
- 5.4: How to correctly add a reference to your code.
- 5.5: Add a reference to your Myself program and redo step 4.6.

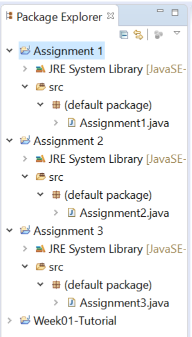

Create Harvard, APA & MLA citations for your bibliography
To start, choose a citation style:
Harvard
APA
MLA
··· More
Change style
powered by CSL
Popular
AMA
APA (6th edition)
Chicago (17th edition, author-date)
Harvard
IEEE
ISO 690
MHRA (3rd edition)
MLA (7th edition)
MLA (8th edition)
OSCOLA
Turabian (9th edition)
Vancouver
Search for...
Academy of Management Review
ACM SIG Proceedings ("et al." for 15+ authors)
ACM SIG Proceedings ("et al." for 3+ authors)
ACM SIGCHI Proceedings
ACM SIGCHI Proceedings - Extended Abstract Format
ACM SIGGRAPH
ACS Nano
Acta Anaesthesiologica Scandinavica
Acta Materialia
Acta Naturae
Acta Neurochirurgica
Acta Ophthalmologica
Acta Palaeontologica Polonica

6. In this exercise, you must perform *dummy* submissions for assignments 1, 2 and 3 by following the steps below. Note: You do not need to write any code for these dummy submissions.

6.1: Create 3 separate Eclipse projects for the 3 assignments. E.g. Assignment 1, Assignment 2, Assignment 3:



6.2: For each project, create a new Java class. E.g. Assignment1.java, Assignment2.java and Assignment3.java as shown:



Tip for later: Assignment 1 requires you to name your .java file a specific way for the actual/final submission. To correctly rename an existing .java file, do a context select (right click) on the .java file name→Refactor→Rename. Do not rename the files from outside of Eclipse as this could lead to issues.

6.3: The Canvas submission system allows you to submit your work an unlimited number of times (only the last submission is official) and it can also be used as a way to backup your work. You must submit the three .java files that you’ve created above as follows (your file names may be different):

Assignment**1**.java must be submitted to Canvas→Assignments→Assignment **1**.
Assignment**2**.java must be submitted to Canvas→Assignments→Assignment **2**.
Assignment**3**.java must be submitted to Canvas→Assignments→Assignment **3**.

E.g. get the **.java** file of Assingment 1 from Eclipse workspace folder→Assignment 1→src.

Common mistake: Students mistakenly submit the dummy assignments via Canvas→Assignments→Tutorial 1.

7. Reading resources.

You may need to refer to the following sources for detailed explanations of topics that are covered.

7.1: Bookmark the Java API Specification:
https://docs.oracle.com/javase/10/docs/api/index.html?overview-summary.html

7.2: Bookmark The Oracle Java Tutorials:
https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html
Note: The ‘language basics’ section would be sufficient for the first several weeks.

8. 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.

9. 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