

Lab report/code submission instructions

Lab Report

Submission contents

All reports must be submitted through the MyCourses system. A submission should consist of a single pdf file with the following naming structure:

Group#_Lab#_LastName1_LastName2.pdf

For example:

Mark Rubin and Catherine Smith are in group 1 and are submitting lab 1. Their report would have the following name: Group1_Lab1_Rubin_Smith.pdf

Outline for a report

The lab report should have the following format and section headings. Reports should have a heading with student names, student numbers, group number. There should also be a meaningful title, course code number, and page numbers. The following section headings should be used:

- Design evaluation
 - Short description of your design (as long as it is concise)
 - Briefly describe your workflow (set of tasks performed to complete the lab)
 - Add a photo for the hardware design (actual or LDD model)
 - Add visuals for the software design (flowcharts, class diagrams)
 - Hand drawn diagrams are NOT accepted
 - For more information about useful software, refer to the document available on MyCourses: [ECSE211 Additional Resources.pdf](#).
- Test data
 - Tables including all the collected test measurements
- Test analysis
 - Tables of resulting test analysis calculations
 - Any sample calculations
 - Discussion of test results
- Observations and conclusions
 - Repeat the question from the lab handout in the report
 - Answer the questions stated in the lab report
- Further improvements
 - Repeat the question from the lab handout in the report



© Instructor and Teaching Assistant generated course materials (e.g., handouts, notes, summaries, assignments, exam questions, etc.) are protected by law and may not be copied or distributed in any form or in any medium without explicit permission of the instructor. Note that infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures.

- Answer the questions stated in the lab report

Data representation

Data collected during the lab, such as distance measurements, must be presented in a table.

- Include units on all measures
- Respect decimal unit precision
 - We don't want to see values like 0.9123222105
 - When performing calculations, remember that the precision of the input also applies to the results e.g. $0.9 + 0.01 = 0.9$
- Consider error margins and the precision of your measurements e.g. how precise is your ruler?

Equations in reports

All equations must be correctly formatted, using LaTeX or equivalent equation editor. In MS Word, you can use the built-in equation editor to generate well formatted equations.

For example:

Represent the equation for calculating the area of a circle.

Good:

$$A = \pi r^2$$

Bad:

$$A = \text{pi} * r^2$$

Explanations

When answering questions or providing explanations, be sure to be concise and only answer what was asked. Furthermore, proper grammar and spelling will be evaluated. It is also important to remain professional and use formal language when writing. Avoid the use of contractions, slang terms, and informal language.

Captions

Figures, tables and photos should all be properly captioned. They should also be referenced in the text where appropriate. For example, you might say "As shown in Fig. 1, We chose to mount the ultrasonic sensor vertically because..."



© Instructor and Teaching Assistant generated course materials (e.g., handouts, notes, summaries, assignments, exam questions, etc.) are protected by law and may not be copied or distributed in any form or in any medium without explicit permission of the instructor. Note that infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures.

Lab Code

Submission contents

All lab code must be submitted through the MyCourses system. A submission should consist of a single zip file with the following naming structure:

Group#_Lab#_LastName1_LastName2.zip

For example:

Mark Rubin and Catherine Smith are in group 1 and are submitting lab 1. Their zip folder would have the following name: Group1_Lab1_Rubin_Smith.zip

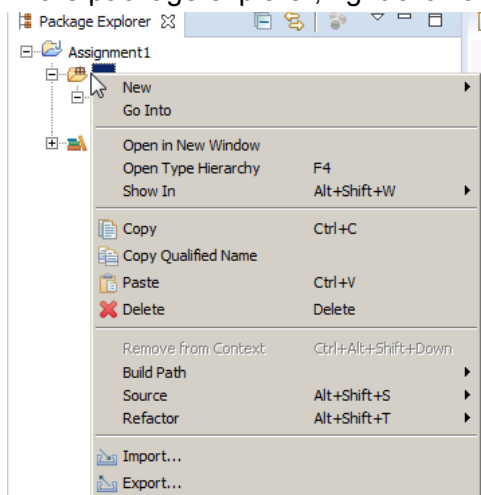
Source files

Submitted source files are expected to compile and run without error.

When creating source files, they should be placed inside a correctly named java package. The proper naming to follow is “ca.mcgill.ecse211.lab#” where # is replaced by the corresponding lab number.

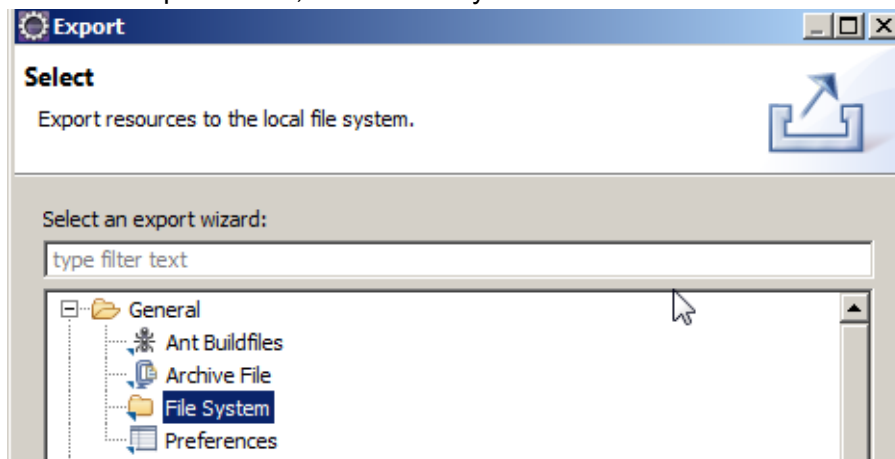
To export source files in their correct package structure from eclipse, perform the following:

1. In the package explorer, right click on the src folder and select export

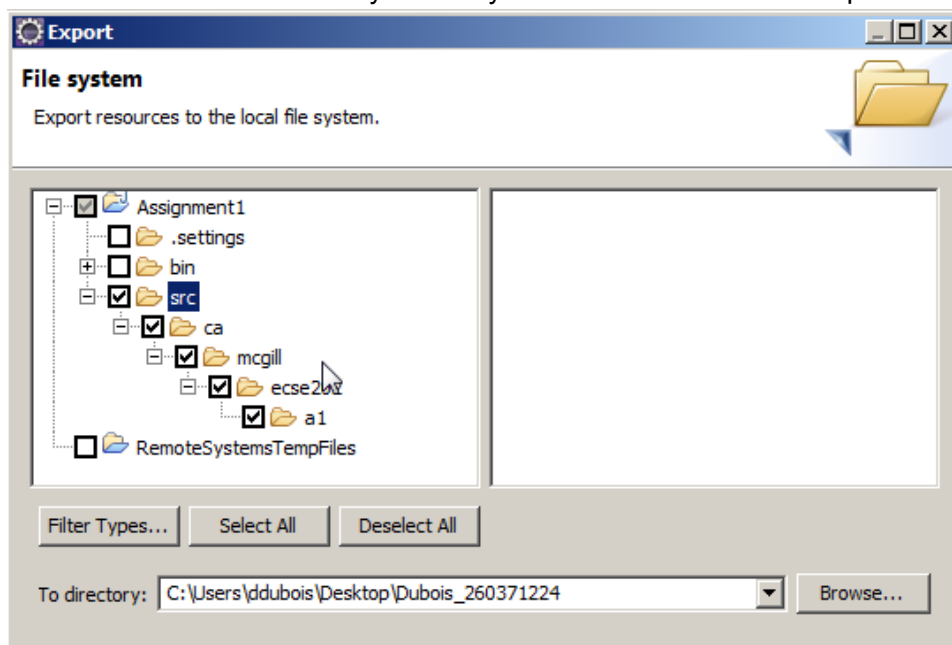


© Instructor and Teaching Assistant generated course materials (e.g., handouts, notes, summaries, assignments, exam questions, etc.) are protected by law and may not be copied or distributed in any form or in any medium without explicit permission of the instructor. Note that infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures.

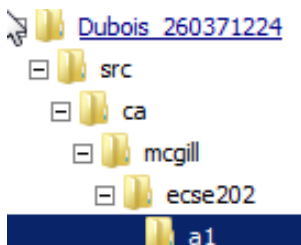
2. From the export menu, select File System



3. Select a destination directory and only select the src folder for export



4. Press finish and the files/folders are exported to the directory with the correct structure



© Instructor and Teaching Assistant generated course materials (e.g., handouts, notes, summaries, assignments, exam questions, etc.) are protected by law and may not be copied or distributed in any form or in any medium without explicit permission of the instructor. Note that infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures.

Zip folder contents

The zip folder must contain the following:

1. *.java source files in their correct package structures, which includes all folders
 - a. ca/mcgill/ecse211/lab#/*.java

Note, this zip does not contain any *.class or *.prj, *.project, or other files. Only submit what is requested for the lab.

Javadoc Commenting

Comments are expected to explain any areas of the code that would not be self-explanatory to someone not taking the course. In other words, we expect to be able to read your code and understand how it works without much difficulty.

In addition to normal comments, we expect basic Javadoc style to be followed i.e. every public class and method should be annotated with what it does and what inputs are required.

For example:

```
package ca.mcgill.ecse211.lab1;

/**
 * This class implements the Wall Follower for Lab1 on the EV3 platform.
 *
 * @author Michael Smith
 */
public class Lab1 {

    /**
     * This main method runs at start, creates main menu, initializes
     * sensors and motors and starts the bang-bang or p-controller based
     * on user button input.
     *
     * @param args Not used.
     */
    public static void main(String[] args) {

        // (Some explanation for next line)
        someclass.doSomething();
    }
}
```

Source code style and formatting

Submitted source code must be well formatted, easy to understand, and follow the [Google Java Style Guide](#). Well formatted code means there is no strange spacing, new lines, or garbage characters. Also, it should be well commented and use meaningful variable names. The



© Instructor and Teaching Assistant generated course materials (e.g., handouts, notes, summaries, assignments, exam questions, etc.) are protected by law and may not be copied or distributed in any form or in any medium without explicit permission of the instructor. Note that infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures.

indentation style, bracket format, etc can be automatically enforced in Eclipse using the following xml file. Download the eclipse-java-google-style.xml file from <http://code.google.com/p/google-styleguide/repo>. In Eclipse, under Window/Preferences select Java/Code Style/Formatter. Import the settings file by selecting Import. Apply the formatter by selecting all the text in a source file, going to the Source menu, and choosing format.

For example:

Good:

The below code is well formatted, uses proper variable names, follows the style guide and is readable. It also uses the correct spacing and indentation. Note, however, that it does *not* contain the *required* Javadoc comments in order to emphasize the difference between good and bad formatting.

```
package ca.mcgill.ecse211.lab1;

public class Lab1 {

    public static void main(String[] args) {

        String myName = "Dirk Dubois"; // My name to be displayed on the console
        System.out.println(myName); // Outputs my name to the console

    }

}
```

The below code is poorly formatted, uses variables names that violate the style guide and is not very readable. The class name is incorrect, the indentation is incorrect, variable names are hard to understand and do not follow the naming conventions, and finally the brackets are incorrect. It also does not have any code comments to help explain its objective.

Bad:

```
package lab1;

public class writes_name_to_console {

    public static void main(String[] args) {

        String B10h3a = "Dirk Dubois";
        String blah2 = B10h3a;

        System.out.println(blah2);    }
}
```



© Instructor and Teaching Assistant generated course materials (e.g., handouts, notes, summaries, assignments, exam questions, etc.) are protected by law and may not be copied or distributed in any form or in any medium without explicit permission of the instructor. Note that infringements of copyright can be subject to follow up by the University under the Code of Student Conduct and Disciplinary Procedures.