Assignment submission instructions

ECSE 420: Parallel Computing

Submission contents

All assignments must be submitted through *myCourses*. A submission should consist of a single .zip file with the following naming structure:

Student1LastName_ Student1Id_Student2LastName_ Student2Id_Assignment#.zip

For example:
Mark Rubin SID 123456789 and Catherine Smith SID 987654321 are submitting assignment 1. There zip folder would have the following name:
Rubin_123456789_Smith_987654321_Assignment1.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.ecse420.a#” where # is replaced by the corresponding assignment number.

Zip folder contents

The zip folder must contain the following:

1. A report (in .pdf format) answering the questions in the assignment
2. *.java source files in their correct package structures, which includes all folders
    a. ca/mcgill/ecse420/a#/*.java
3. Any input files related to the assignment

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

Report PDF file

The report pdf must contain your groups’ names and student IDs. The pdf should answer all the questions asked in the assignment. For code related questions, it should refer to the appropriate source files and include any input and output passed or generated by the code. The report should be well-formatted and easy to read. Please be explicit about which question you are answering. The PDF file must have an appendix that contains all the source code. This can be used for referencing code when answering questions. Note, you still must submit the source files as well.

Source code style and formatting

Submitted source code must be well formatted, easy to understand, and should 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.

For example:

Good:

```
package ca.mcgill.ecse420.a1;

public class Question1 {

  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 above code is well formatted, uses proper variable names, follows the style guide and is readable. It also uses the correct spacing and indentation.

Bad:

```
package ca.mcgill.ecse420.a1;

public class writes_name_to_console {

  public static void main(String[] args) {

                      String Bl0h3a = "Dirk Dubois";
    String blah2 = Bl0h3a;


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

The above 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.