Assignment Requirements Information

You are required to submit your work **electronically** on Moodle. Please note that not following the submission format outlined below will result in **penalties**. Late submissions are graded with **0.25%** penalty per minute after the deadline.

1.1 Submission Format

Your electronic submission as a zip file must include all of the **Java** files that you wrote/created. If a task/problem contains **red label**, you should use that particular label as the **name of your class and the file containing that class**. All the requirements of the task should be implemented **inside the labeled class**. **DO NOT** put all your codes in text/word/pdf files.

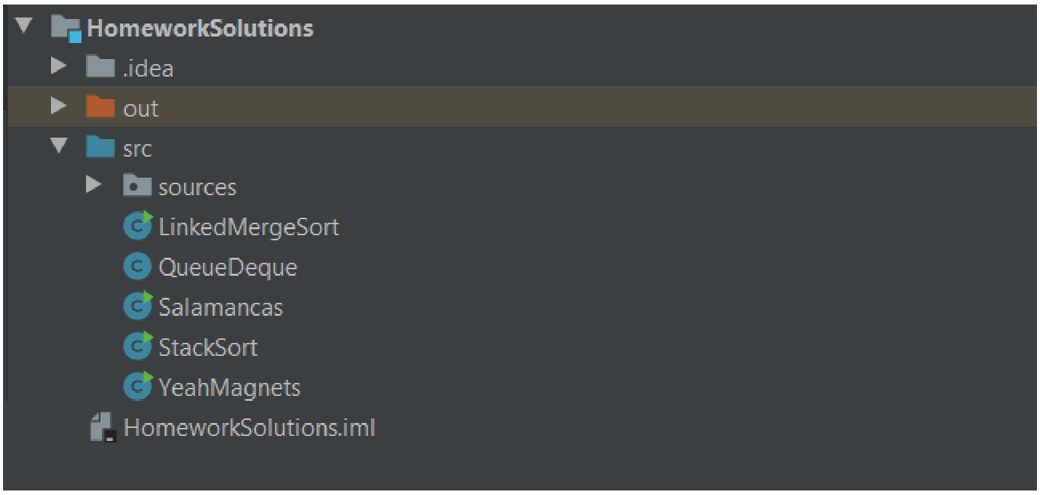
If the problem needs you to use/build upon a data structure covered in class, **use the source codes provided** on Moodle (**sources folder**) to do that. Create a subfolder named **sources** inside your main folder and put all of the source codes needed to run your program there.

Follow the steps outlined below to ensure that your submission is correct:

1. Create a new project for the current homework (if you are using an IDE) or a new folder (if you are compiling using javac).
2. Make sure all of your solutions are in the **default package** (src in case of an IDE). Thus, none of your solution files should have any package name at the top of the file.

   Make sure that there is **NO** long nesting of folders inside folders that is often automatically generated by IDEs (e.g., net.com.net.net.ineverdomyhwsontime.net.com.com.scr).
3. Inside your default package, create a folder named ``***sources***’’ and put all the relevant codes from ``sources’’ folder on Moodle here (make **NO** changes to any of these files). When you need to use any of these classes, import them from the sources package inside your solution using *import sources.*;* statement. In general, DO not use any existing java classes/interfaces (e.g., java.util.Collections) unless explicitly allowed by the problem formulation. The exception to this rule are the interfaces *Iterator, Iterable, Comparator and Comparable*.

The picture below shows how a project setup for the first homework should look in IntelliJ:



```
▼ HomeworkSolutions
  ▶ .idea
  ▶ out
  ▼ src
    ▶ sources
      LinkedMergeSort
      QueueDeque
      Salamancas
      StackSort
      YeahMagnets
    HomeworkSolutions.iml
```

4. When submitting your work, ONLY submit the files from the outer (src) folder. Do no submit the source folder (we are going to assume that all your codes should be compatible with the sources folder on Moodle). ZIP all the files (NO RARs) and upload the zip on Moodle.

1.2 Program Coding and Commenting

We expect you to use meaningful variable names, a consistent indenting scheme, and potential comments. Besides header comments, you may have normal commenting within methods to document local variables and any computations that aren't obvious. Below are some guidelines as to what is desirable to include in header comments:

- Class headers: This is the header comments at the top of a file containing a class definition.
  This header should contain the following information:
  - Outline of the purpose of the class

- Function headers: This is the header at the beginning of any function definition.
  These headers should contain the following information:
  - Outline of the purpose of the function

Do not use any OS-specific functions in your code. Programming problems are generic, so make your code generic as well.