For this homework assignment you will produce a design for the second programming project. You must produce a high-level representation of your design, in the form of a class diagram. The class diagram must include all the classes you will use, and indicate the relationships among them (or of the corresponding objects, if it's easier to think of it that way). Use the notations given in the course notes to indicate association and aggregation relationships. For each such relationship, you should indicate the multiplicities involved. For association relationships, be sure to provide a descriptive name.

The class diagram does not need to indicate members of classes, just class names. Note that the array template is not to be represented in the class diagram, but that instantiations of it will be. For convenience, name those instantiations using standard template notation, such as `Array<Book>`.

You must also produce a class form for each class, and also for the array template. Use the same approach for the class diagrams that was given in the `Tank` design homework, with one extension. For the private members, add an indication that they are, in fact, private. The notation you use is up to you, but any member not clearly specified as private will be assumed to be public. Operation forms are not required.

The first thing you need to do is read the specification for that project carefully. Pay particular attention to the section that discusses the overall system design and goals.

You should apply the design techniques presented in the course notes and lectures, and apply the presented design evaluation criteria to your resulting design. The process of design is not linear; you may need to revisit some of the design decisions after evaluating your work, and then re-evaluate the revised design.

There are probably many reasonably good designs for this project, and certainly a number of poor ones. A design is not good simply because it is possible to produce a working implementation from it.

You may decide that you want to make use of additional classes, not identified in the project specification. That is allowed, but not required. If you decide to implement additional classes, you must include them in the class diagram and provide class forms for them as well.

What to turn in and how:

Your design must be submitted in a format that can be opened and edited (to insert comments from the evaluator) in MS Word. An MS Word document is acceptable, as is a plain text file, neatly formatted. The simplest way to draw the class diagram is to use the drawing tools in MS Word. You may use another approach if you like, but bit-mapped drawings (gif, jpeg, bmp, etc.) are not acceptable.

Submit the file containing your design to the Curator System; the assignment will be listed as `P2Design`. The file should be in a format that can be read in MS Word. Do not zip the file. Instructions for submitting to the Curator are given in the *Student Guide* at the Curator website:

`http://ei.cs.vt.edu/~eags/Curator.html`.

Be sure to follow those instructions carefully. You will submit your design via the URL:

`http://spasm.cs.vt.edu:8080/curator/`

You will be allowed to submit your solution up to three times. Your last submission will be scored. Note: this assignment will be graded by the GTAs, not by the Curator system. You will not be receiving email from the Curator containing your score, although you will probably receive a confirmation message. You should confirm your submission, including its size, on your Curator Home page.