

For this homework assignment you will produce a design for the Tank class required for the first programming project. 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 the Tank class, 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, which will be used by the Tank class, you must include class and operation forms for those as well.

## Design Representation

Create a class form to represent the Tank class interface, and a separate operation form for each proposed member function. Class and operation forms are simply tables containing design documentation that will provide a quick overview of the final design. They do not provide a formal record of the deliberations that occurred during the design process. Class and operation forms should not include implementation-level decisions, such as variable and parameter types. Sample templates for class and operation forms are given at the end of this assignment.

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.

## What to turn in and how:

Submit the file containing your design to the Curator System. 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.

**Sample class description form:**

Class Name	Complex
Purpose	represent a complex number and provide associated operations
States	none
Constructors	Complex(Real = 0.0, Imaginary = 0.0)
Operations	
Mutators	setReal, setImaginary, Magnitude, Conjugate, +, -, /, *, ==
Accessors	getReal, getImaginary
Fields	Real Imaginary

**Sample operation description form:**

Prototype	Complex operator+(Complex rightOperand)
Purpose	to compute the sum of two Complex objects
Data received	right operand of binary expression, left operand is the execution scope of the operation
Data returned	Complex object representing the sum of the objects
Remarks	Overloaded addition operator is provided to support natural coding of expressions involving Complex objects.

An MS Word file containing blank copies of these is posted on the course website. Feel free to use these if you wish.