Advanced Programming Using Java - Lesson 3 (Chapter 2 Part A)

GOALS: Learn about and use objects, classes, encapsulation, and references, private keyword

Make a naval battle simulator where Weapon Strength is the maximum damage a weapon can do, and Armor Strength subtracted from that random range to get damage (capped to only 0 or positive values so weapons don’t ever repair a ship) which is subtracted from Hit Points. Parrying doubles the difficulty in damaging an object.

* Open NetBeans
* File, New Project
	+ Java, Java Application, Next
	+ Name it “acp2”, uncheck “Create Main Class”, Finish
	+ File, New File, Swing GUI Forms, JFrame Form, Next
		- Name it MainForm
		- Name the package “acp” then click Finish
* Click Run. When asked, select the main class acp.MainForm then click OK to set it as the main class. After packaging is finished, you will see a blank program, and you can close it.
* Create a form for a ship battle simulator:



* + In the Properties tab, change the Text of the labels and buttons and TextFields
	+ In the Code tab on the right panel, change the Variable Names of the TextFields to ship1AS, ship1WS, ship1HP; the parry buttons as ship1AttackButton and ship1ParryButton; and so on (same for ship2 widgets except with 2).
* Make the form taller, drag a list to it
	+ In Code tab on right, change Variable Name to logList
	+ In Properties tab on right, by “model” click “…” then delete all of the text from the list
* Create a ship class that can be used for both ships:
	+ File, New File, Java, Java Class, Next
	+ Name it “Vehicle”
	+ Make sure package is acp, then click Finish
* In the Vehicle class, create variables for the situation:



* + Highlight each variable name (such as by double clicking the name) then wait for the option to bulb to appear then click “move initializer to constructor” (if you can’t see the menu, try Alt-Enter while variable name is highlighted) for each one.

We can control the way the variables are used, so that the data isn’t skewed. We can make accessors (“getters”) and mutators (“setters”) automatically using the Refactor feature of NetBeans, leave the variables private, then later we can make custom functions that only mutate the object (manipulate the values of the member variables) according to certain rules.

* Highlight each variable name (except “displayName”) then:
	+ Right-click, “Refactor,” “Encapsulate Fields…”
	+ Click “Refactor” button
	+ Repeat for each variable