Operation **J**ava Blitz

Designing Classes:

NCC 2012 Problem Statement Solution Approach

Dr. Nannette Napier

1) Review the Problem

- You are to build a "library" – a set of classes that can be used to accomplish related tasks

- Spend 5 minutes reviewing the project description
  - Focus on the section called "Requirements"
  - On the back, look at the "Grading Criteria" and "Optional Enhancements"

2) Write about the Problem

- On your scratch paper, jot down
  - What are possible class names?
  - What kind of input or data do you need to store in your class? *(Instance variables)*
  - What kind of behaviors or action is required with the data? *(Instance methods)*

ROUND 1

Class Name: Length

- Data to store (Instance variables)
  - Number of miles, double
  - Number of yards, double
  - Number of feet, double
  - Number of inches, double
- Constructors
  - No-arg constructor
  - 4-arg constructor (miles, yards, feet, inches)
- Generate the getters and setters
- Behaviors
  - **representAsMiles:** Convert any length to miles *(return an exact number of miles that this length represents as a double)*

  +representAsMiles():double
  - **representAsYards:** Convert any length to yards *(return an exact number of yards that this length represents as a double)*
  - **representAsFeet:** Convert any length to feet *(return an exact number of feet that this length represents as a double)*
  - **representAsInches:** Convert any length to inches *(return an exact number of inches that this length represents as a double)*
  - The **toString()** method will return the "common measurement" as illustrated in the requirements
  - **plus** method – adds a length to this one
    +plus(anotherLength:Length):Length
  - **minus** method – subtract one length from this one
    +minus(anotherLength:Length):Length

Open LucidChart and
Create the UML Class
diagram for Length

LengthTester: Has main method
- Create 3 Length objects
- SOP each object
- Add length1 plus length2. SOP the answer
- Subtract length2 from length3. SOP the answer

Class Name:

- Data to store
- Behaviors

What I don't like about this one…

ROUND 2

Class Name:

- Data to store

- Behaviors

Class Name:

- Data to store

- Behaviors

Sample library usage: Round 2

// Library must be configurable to take in
// and return exact measurements
Length length1 = new Length();
length1.addMiles(5.17045455);

Unit	Divisions	SI Equivalent
Exact relationships shown in **boldface**		
International		
1 *point* (p)		352.777 778 µm
1 *pica* (P/)	**12 p**	4.233 333 mm
1 *inch* (in)	**6 P/**	**25.4 mm**
1 *foot* (ft)	**12 in**	**0.304 8 m**[9]
1 *yard* (yd)	**3 ft**	**0.914 4 m**[9]
1 *mile* (mi)	**5 280 ft** or **1 760 yd**	**1.609 344 km**
US Survey		
1 link (li)	$^{33}/_{50}$ **ft** or 7.92 in	0.201 2 m
1 (survey) foot (ft)	$^{1200}/_{3937}$ **m**	0.304 800 61 m[9]
1 rod (rd)	**25 li** or **16.5 ft**	5.029 21 m
1 chain (ch)	**4 rd**	20.116 84 m
1 furlong (fur)	**10 ch**	201.168 4 m
1 survey (or statute) mile (mi)	**8 fur**	1.609 347 km[9]
1 league (lea)	**3 mi**	4.828 042 km
International Nautical[9]		
1 fathom (ftm)	**2 yd**	**1.828 8 m**
1 cable (cb)	**120 ftm** or 1.091 fur	**219.456 m**
1 nautical mile (NM or nmi)	8.439 cb or 1.151 mi	**1.852 km**

Liquid volume		
Most common measures shown in *italic* font		
Exact conversions in **bold** font		
Unit	Divisions	SI Equivalent
1 minim (min)	~1 drop or 0.95 grain of water	61.611 519 921 875 µL
1 US fluid dram (fl dr)	**60 min**	3.696 691 195 312 5 mL
1 *teaspoon* (tsp)	**80 min**	4.928 921 593 75 mL
1 *tablespoon* (Tbsp)	**3 tsp** or **4 fl dr**	14.786 764 781 25 mL
1 *US fluid ounce* (fl oz)	**2 Tbsp** or 1.0408 oz av of water	29.573 529 562 5 mL
1 US shot (jig)	**3 Tbsp**	44.360 294 343 75 mL
1 US gill (gi)	**4 fl oz**	118.294 118 25 mL
1 *US cup* (cp)	**2 gi** or **8 fl oz**	236.588 2365 mL
1 (liquid) *US pint* (pt)	**2 cp** or 16.65 oz av of water	473.176 473 mL
1 (liquid) *US quart* (qt)	**2 pt**	0.946 352 946 L
1 (liquid) *US gallon* (gal)	**4 qt** or **231 cu in**	3.785 411 784 L
1 (liquid) barrel (bbl)	**31.5 gal** or $^1/_2$ **hogshead**	119.240 471 196 L
1 oil barrel (bbl)	**42 gal** or $^2/_3$ **hogshead**	158.987 294 928 L
1 hogshead	**63 gal** or **8.421 875 cu ft** or 524.7 lb of water	238.480 942 392 L

Caveats

- There is always more than one way to solve a problem
- The 2014 NCC problem may be nothing like this
- Principles I consider with my approach
  - Time spent thinking about the problem is well worth it, but we don't have all day!
  - Solution should demonstrate OO knowledge (inheritance, interfaces, abstract classes)
  - Solution should work!
  - Design to take advantage of 2 person team

Would also consider using

- The Builder Design Pattern. Read more here: http://en.wikipedia.org/wiki/Builder_pattern

- Familiarity with design patterns can make coming up with an approach a lot easier. This is a great topic for self-study

Resources

- Practice Problems
  - http://www.cs.utexas.edu/users/scottm/uil/practiceMaterials.html
- Advice
  - http://programmers.stackexchange.com/questions/109207/how-to-prepare-for-a-programming-competition-graphs-stacks-trees-oh-my
  - http://www.quora.com/topcoder/How-should-I-practice-so-that-I-will-be-at-a-level-where-I-can-approach-TopCoders-Div1-500-problems-with-confidence