Application Design Using Java

Lecture 03

Scripts

- Windows
  - batch (.bat) files
  - command (.cmd) files
  - See https://stackoverflow.com/questions/148968/windows-batch-files-bat-vs-cmd
  - http://steve-jansen.github.io/guides/windows-batch-scripting/
- *nix (including MacOS)
  - shell scripts (e.g., bash)
  - https://itnext.io/bash-scripting-everything-you-need-to-know-about-bash-shell-programming-cd08595f2fba
  - https://devhints.io/bash
- Automation with scripts (e.g., testing)

Javadoc

- https://www.oracle.com/technical-resources/articles/java/javadoc-tool.html
- javadoc

Coding Conventions

- https://google.github.io/styleguide/javaguide.html

Primitive Types

- Whole (integer) numbers
  - long (l – not good!, L)
  - int
  - short
  - byte
  - Prefixes:
    - 0x or 0X hex
    - 0 octal
    - 0b or 0B binary

Type	Storage Requirement	Range (Inclusive)
int	4 bytes	–2,147,483,648 to 2,147,483, 647 (just over 2 billion)
short	2 bytes	–32,768 to 32,767
long	8 bytes	–9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
byte	1 byte	–128 to 127

- Integer.MIN_VALUE, Integer.MAX_VALUE, similar for other integer types

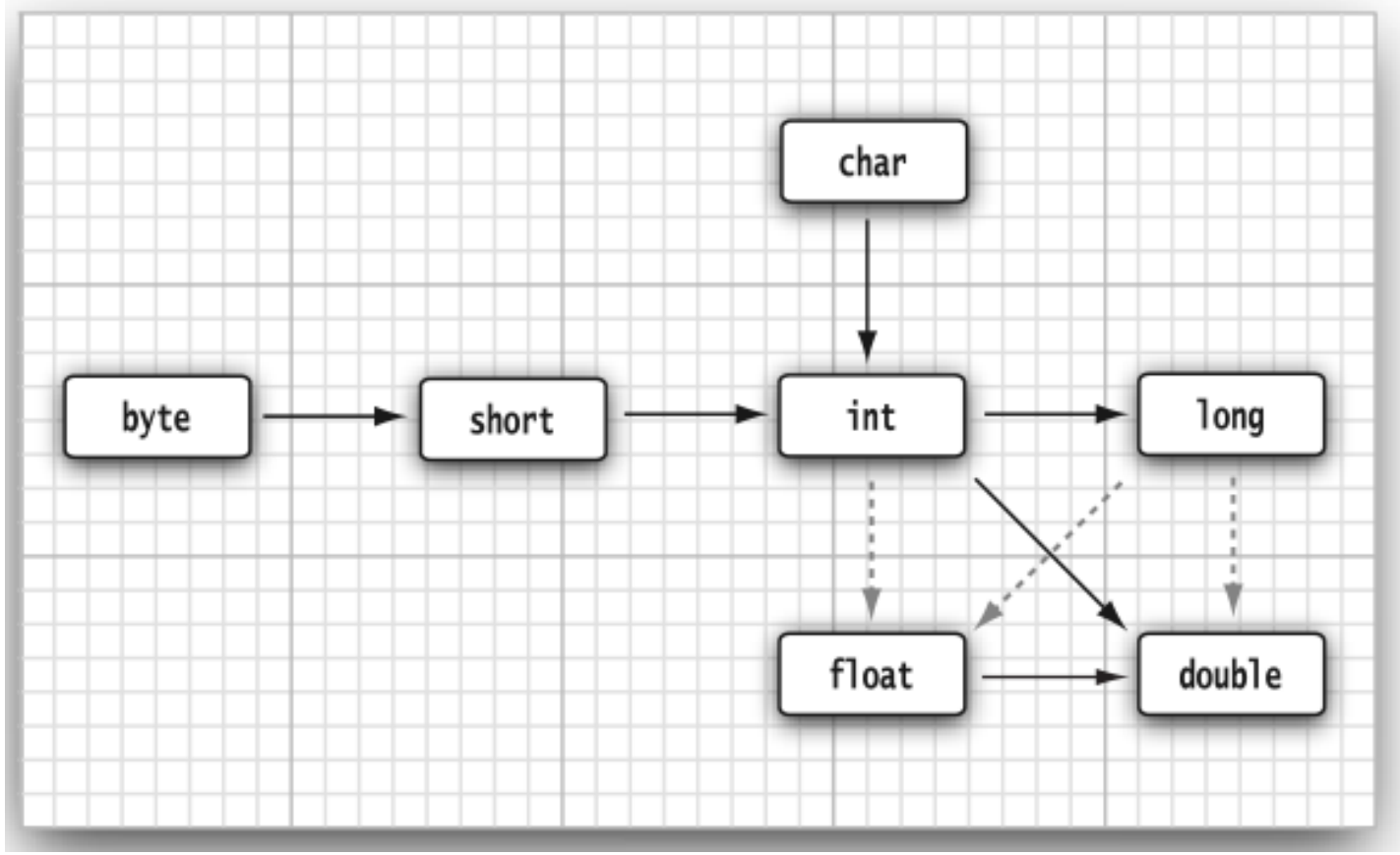
Primitive Types

Type	Storage Requirement	Range
float	4 bytes	Approximately ±3.40282347E+38F (6–7 significant decimal digits)
double	8 bytes	Approximately ±1.79769313486231570E+308 (15 significant decimal digits)

- Fractions
  - double (d, D)
  - float
  - Literals can be hexadecimal:
    0.125= 2^{-3} => 0x1.0p-3
  - Float or Double.POSITIVE_INFINITY, Double.NEGATIVE_INFINITY, and Double.NaN
  - NaN is tricky:
    - x == Double.NaN // is never true
    - Use Double.isNaN(x)
  - Floating-point numbers are *not* exact!

Primitive Types

- Numeric Conversions



//TODO before next lecture:

- Practice problems for Lecture 3
- Challenge problem (can work in teams):
  - Compare the speed of performing different arithmetic operations (addition, multiplication, division, etc.) for different numeric data types. Post on Webex Teams or Submitty Forum.