Java By Abstraction - Companion Notes

Topic 3 - Application Programming Interface (APIs)

CSE 1720, Winter 2010, Version 1.0, Prepared by: M. Baljko

see first Companion Notes for Legend JD, PT, Ex, etc

Section 3.1; JD 3.1-3.2; PT 3.1-3.2; Ex 3.1-3.4

- 3.1 – the API for Java can be found at:
  `http://java.sun.com/j2se/1.5.0/docs/api/`
- JD 3.1 – read
- JD 3.2 – can ignore this (unless you are super-keen)
- PT 3.1 primitive types – passing by value LEAVE ASIDE FOR NOW; we will talk about this later
- PT 3.2 makes mention of the notion of promotion among numeric types. We have deferred our discussion of the primitive data types for now (which include the numeric types (`short`, `byte`, `int`, `long`, `float`, `double`, and `char` (yes – char!)) and the Boolean type `boolean`). So for now, the take-away point of PT 3.2 is that there are many different methods in the `PrintStream` class, all sharing the same name (but with different signatures – prove this to yourself! go have a look at the Java API)
- Ex 3.1 mentions the `Integer` class – this is an example of a *wrapper* class; it is the nonprimitive version of the primitive `int` type.

Section 3.2; JD 3.3-3.5; PT 3.3-3.4; IMD 3.1-3.2

- 3.2.1 read
- 3.2.2 read, the "Analysis" subsection makes several good points. In the "Design" subsection, one addendum that I would like to add is that the first thing the "design team" should do is to investigate whether there already exist implementations that provide mortgage-calculation services (e.g., do a web search or look in various code repositories). The derivation of a given monthly payment for a set of particular parameter values can be encapsulated and should be delegated to its own module. If such a service is not available (under appropriate terms/conditions), then only then should the design team consider implementing its own module.
- 3.2.3, PT 3.3, IMD 3.1 read this and understand how the arguments to `printf` should be specified
- 3.2.4 partially defer – know that these operators exists and what they do; the part about precedence we will defer until later; PT 3.4 makes an important point about the == operator
- 3.2.5, JD 3.3 read and understand how to invoke the crash method of Toolbox, as well as the alternative technique of throwing the exception directly
- 3.2.6, JD 3.4, JD 3.5, IMD 3.2 understand the idea of assertions; how they work and their purpose; we will use them later

Section 3.3

- REALLY IMPORTANT!!!! Note that this builds on material that was described in the "Alternative Version of Case Study 2.3.4 – Part 1" Also read to "Alternative Version of Case Study 2.3.4 – Part 2"

Case Study 3.3.3

- Read this – we will use this in the labs