Programming Languages: Principles and Paradigms
13.1, 13.3, *13.4, *13.10, 13.18, 13.21ab

1.
code reuse: subtype polymorphism, parametric polymorphism (with generics), object instantiation (making several objects of the same type that all use the same methods)
type safety: Java has built-in error handling for index-out-of-range array references
abstraction: Objects are abstractions based on a class of data and the specific kind of control (methods) associated with it.
encapsulation: classes, packages

3.
class ClosedFigure { ... }
interface Drawable { ... }
class Rectangel extends ClosedFigure implements Drawable { ... }
class Circle extends ClosedFigure implements Drawable { ... }

4.
(a). ?
(b). The STL. Apparently the STL is faster than Collections. Also C++ iterators are more varied and more powerful than Java iterators.

10.
See my cnc_submitted (Clite_final.tar) implementation of Clite, which is purely (as pure as Java can be) object oriented.

18.
Declaration is a client of Type.
Value is a subclass of Expression.

21.
See #10