Packages

Folders in which we create Java classes

“family name” for classes

.Java.util.ArrayList, for example is the “fully-qualified” name for the ArrayList class

.If we want to use it, we must either use the full name to reference it, or import the class at the top of the page. Ex: java.util.\* or import java.util.ArrayList;

.Subpackages have no special connection to parent packages. i.e com.ex.\* will not import com.ex.sub.ClassName

Access modifiers/levels

Determine where, from outside of a class, an entity can be accessed.

Public – accessible everywhere

Protected – accessible within same package & from subclasses

Private-only accessible within the class it is declared in [no modifier] aka “default” or “package” – only accessible within same package.

Variable Scopes

.Block scope

Encapsulation - concealing of access to an objects fields.

* Object fields
* Private instance vars,
* Public getter(accessor methods) & setters(mutator methods)

Inheritance – Ability of objs to share & inherit behavior

Abstraction – hiding implementation details, only exposing functionality

* Abstract classes & interfaces

C1 extends C2

C1 implements I1

I2 extends I1

Polymorphism – Ability for objs to behave in various manners

* Method overloading: same name, different signature (same class)
* Method overriding: same name, different implementation (inherited classes)
* Covariant types ie. Human h = new Adult();