Class Huddle

A quick conversation about what bootcamp is all about.

1) Talk about refactoring

Managing Inheritance

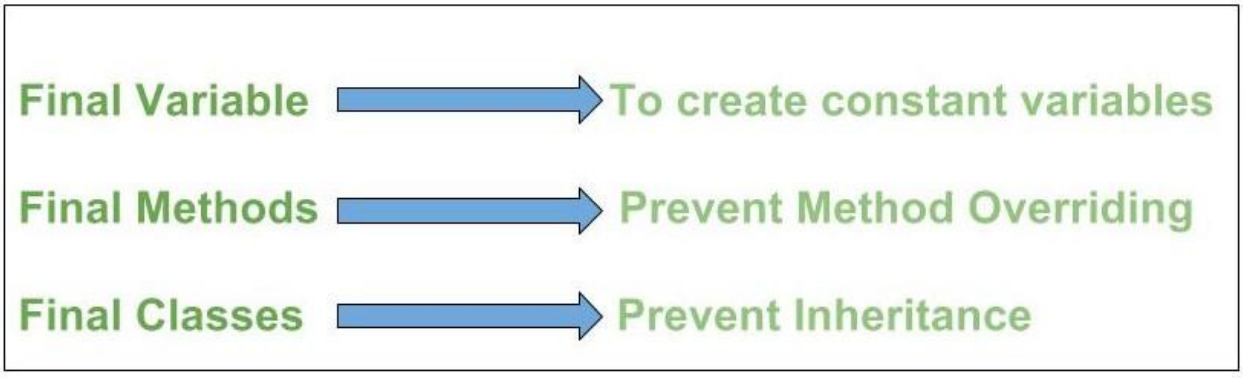
Abstract keyword (“The ghosts of our ancestors compel us.”)

An abstract class cannot be instantiated, so there must be a child class to do the actual work.
An abstract method has no body, so a child class must implement it (like interface method)

Can’t have an abstract method except in an abstract class.

Interfaces should not have full methods, only method signatures.

Final keyword (“Don’t change that!” or “I have the final word!”)



Protected keyword (“Keep it in the family.”)

While a **public** property or method is accessible and a **private** property or method is only available with the class, a **protected** property or method is accessible by the class and any of its descendents. It is also visible to any other classes in the same package., because they are presumed to be related to each other. While the *no modifier* access is defined below, I don’t want you to use it. It will just confuse you.

Simple rule of thumb: all variables declared in a class (referred to as properties) should use **private** or **protected** unless somebody who pays your salary says otherwise.

This chart was in the reading, but is worth saving here as well.

Modifier	Same class?	Same package?	Descendant?	Anywhere?
`public`	Yes	Yes	Yes	Yes
`protected`	Yes	Yes	Yes	No
no modifier	Yes	Yes	No	No
`private`	Yes	No	No	No

Static keyword (“Your one and only”)

A **static** property or method belongs to the class and not the instances, so everyone shares it. Change a static property and it is changed for every instance. A static method therefore can’t use instance properties because it has no concept of the instance. Static methods can only call static methods and only use static properties. Regular methods can access both static and non-static properties and methods.

Resources

[The 3 Pillars of Object-Oriented Programming (OOP) Brought Down to Earth]

[Inheritance & Polymorphism (& smelly butts)]

IS-A relationship for inheritance. Chicken IS-A FarmAnimal

HAS-A relationship for variables. Cat HAS-A nickname

CAN-DO relationship for interfaces. PoppingCorn CAN-DO Sellable