

Day 1:

Decentralized Version Control
→ every body gets a copy!

Github

Centralized
→ only one copy

Day 2:

→ Review for OOP and Classes.

Starting with Java.

method declaration or signature

↳ public static void

↳ access modifiers

public, private, protected

↳ modifiers

static or (omit)

↳ return type

void, int, other...

Day 5:

Application:

Model → Pojo → Plain Old Java Object → Just a collection of information, something to store.
→ Properties

Service → behaviors

Service often deals with the model
Such as Game and GameService

O object
O oriented
P programming

- Abstraction → user/caller doesn't need to know what's going on under the hood.
- Polymorphism → objects can take many forms through method overloading and overriding.
- Inheritance → IS-A relationship, child classes inherit fields/methods from parent class.
- Encapsulation → objects are whole and self-contained, private information is withheld

Inheritance

Interface → does not contain body of methods, and does not have fields.

Classes

- Abstract Class → cannot be instantiated
 - like animal → is it dog? cat?
 - can include default behaviors and have fields.

S
O

Liskov Substitution → subtypes must be substitutable for base type without altering correctness of program

I
D

principles