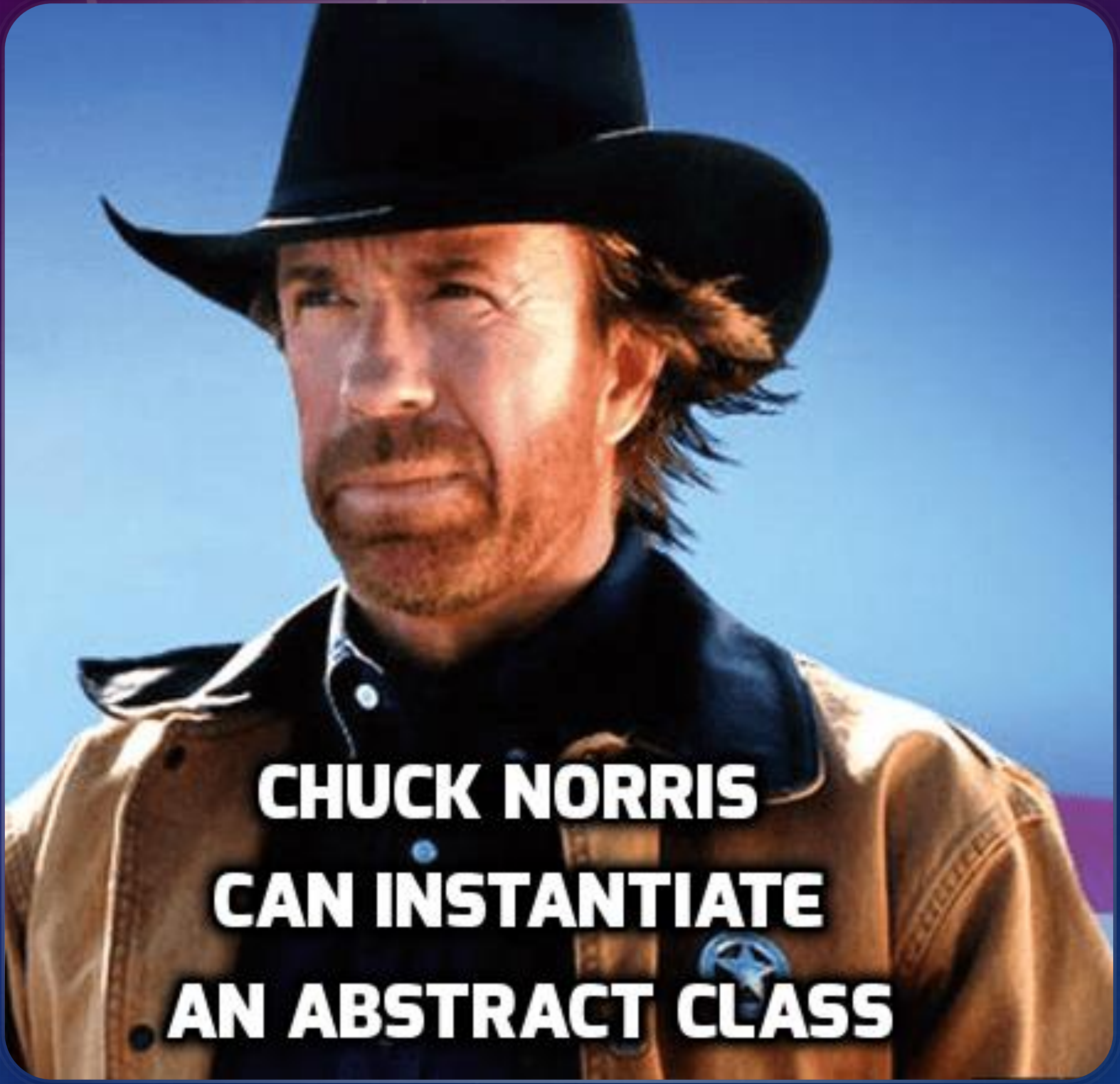
ABSTRACTION

BASED ON THE SLIDES OF

STEPHEN CLYDE PH.D., UTAH STATE UNIVERSITY



REVIEW OF LOCALIZATION OF DESIGN DECISIONS AND ENCAPSULATION

Localization of Design Decisions

- What does it mean?
- Why is it important?
- How do we do it?

Encapsulation

- What does it mean?
- Why is it important?
- How do we do it?

ABSTRACTION

Abstraction: summarizing or generalizing something to focus on the ideas that are most relevant to a conversation or purpose.

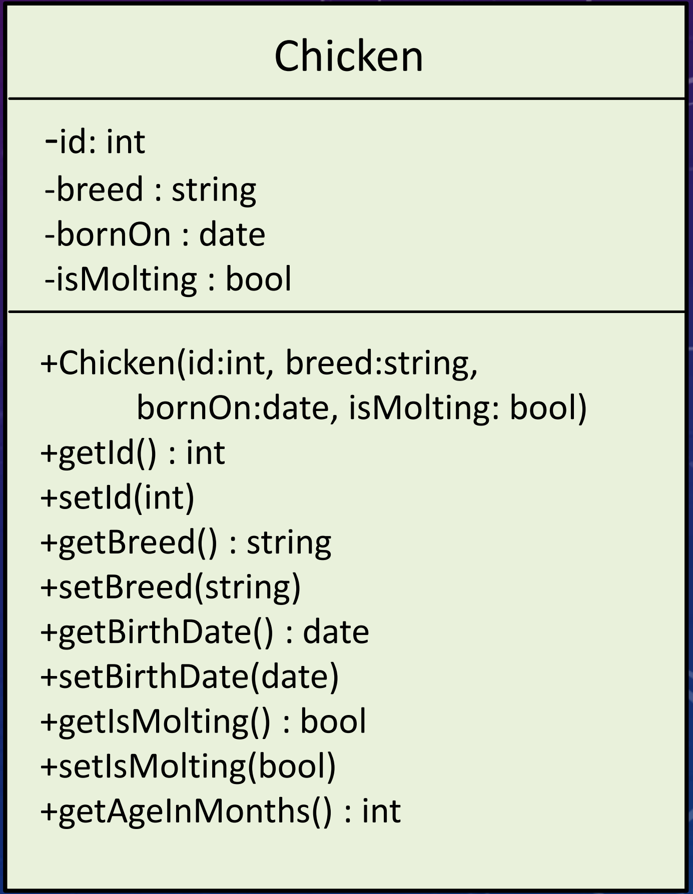
In programming, abstraction (the verb) is the creation of an interface for a component, e.g., a class definition, that exposes certain details necessary for working with that component, while hiding other details

An abstraction (the noun) is a description or definition that leaves out unnecessary details. A class definition is an abstraction

- Public properties, like public methods, exposed or interesting details
- Private properties and the implementation of the methods hide details users of the class don't need to depend on
- Irrelevant details are left out of the class definition all together

EXAMPLE

- Think about chickens in a farm information system
  - What are some properties of chicken (state or behavior)?
  - Which of those are pertinent to the farm information system?
  - The process of sifting through all of the properties and focusing on what is important is abstraction (the process or verb)
  - The resulting class definition for chickens is an abstraction (the noun)
- Note: the Chicken class would be very different if we were building a chicken coop simulator, instead of farm information system



```
Chicken
-id: int
-breed : string
-bornOn : date
-isMolting : bool
+Chicken(id:int, breed:string,
        bornOn:date, isMolting: bool)
+getId() : int
+setId(int)
+getBreed() : string
+setBreed(string)
+getBirthDate() : date
+setBirthDate(date)
+getIsMolting() : bool
+setIsMolting(bool)
+getAgeInMonths() : int
```

ABSTRACT CLASSES AND INTERFACES IN JAVA

- abstract: class, method
  - class: an object of an abstract class can not be instantiated
  - method: the derived class must implement that method
- Interface:
  - Another component of Java and maybe many other languages as a concept
    - interface: inherits from other interface
    - class: a class implements an interface
- Java Keywords: abstract, interface, implements, @override

INTERFACES IN JAVA

1. interface IFarmAnimal { … }
2. IFarmAnimal farmAnimal; //   ?
3. IFarmAnimal farmAnimal = new FarmAnimal(); //?
4. farmAnimal = new Chicken();
5. Collection<Person> persons;
6. persons = new Collection<>(); // ??
7. persons = new ArrayList<>();//??
8. Set<Integer> integers;
9. integers = new Set(); //   ??
10. integers = new HashSet(…);//??

1. int a;
2. System.out.println(a); ?

3. int a = 0;
4. System.out.println(a); ?