

The background of the slide features a light gray circuit board pattern with various traces and circular components. A solid dark gray horizontal band runs across the middle of the image, serving as a background for the text.

CMSC 131

Fall 2018

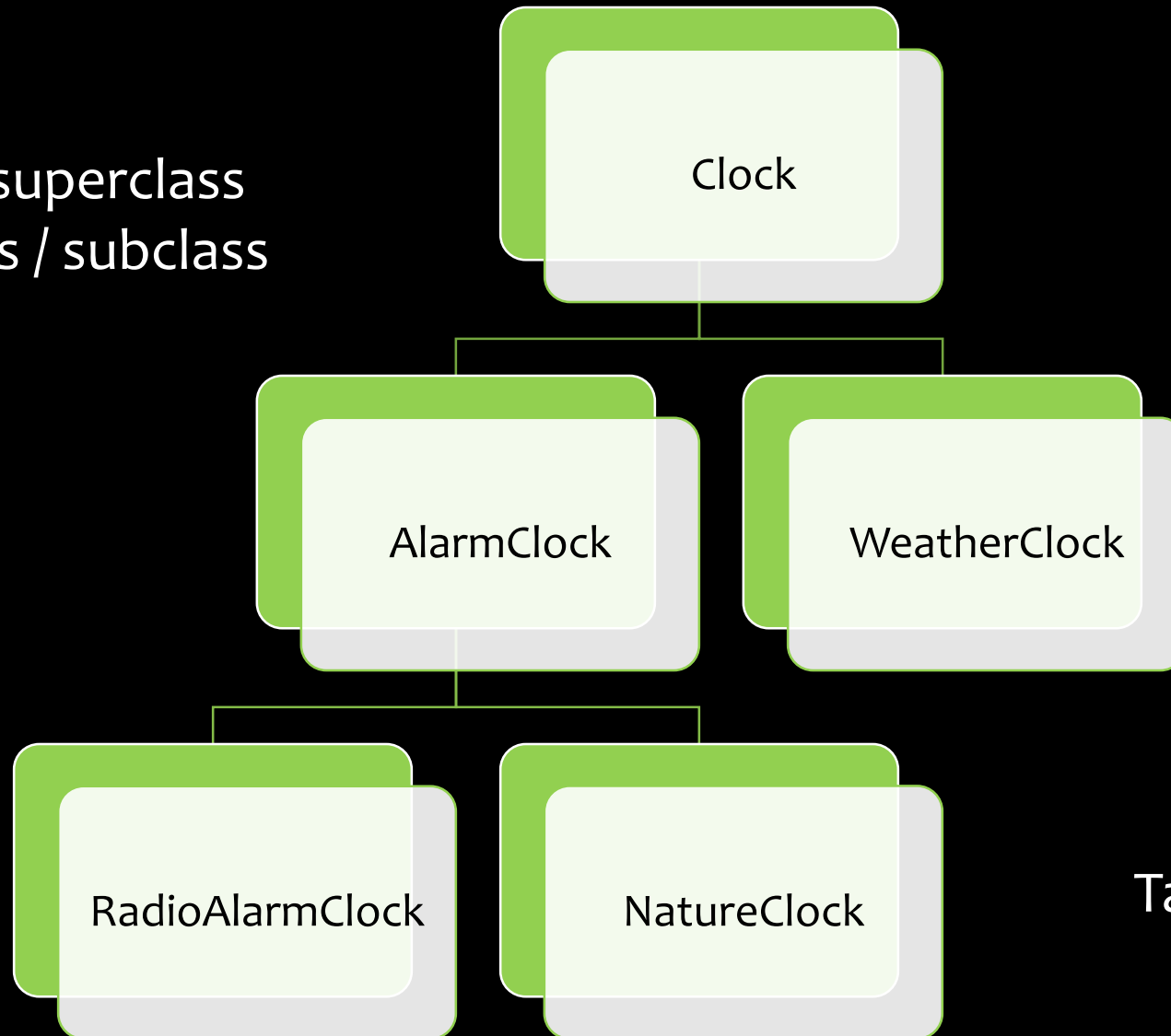
Announcements

- Project #6 has been posted
 - Comments: You should edit them!
- Exam #2 one week from today

Recall: Inheritance Diagram and “IS-A” relation

Vocabulary:

- Base class / superclass
- Derived class / subclass



Talk about “Transitivity”

Polymorphism via Extension

```
void moveForwardOneHour(AlarmClock c) {  
    ...  
}
```

What kinds of objects can be used as arguments?

What methods can be called on c?

Overriding vs. Overloading

What is a method overload?

What is a method override?

Let's demonstrate:

Write a unique implementation of “tick” in the AlarmClock class (override)

Add a different “tick” method (with a parameter) to the AlarmClock class (overload)

Object class

Every class we write is an extension!

If not specified, your class extends “Object”

Let’s look at the API for the Object class

What kinds of objects can be assigned:

Object x = ?

Let’s re-draw the inheritance diagram for clocks

Equals method, revisited

```
public boolean equals(Clock x) {  
    ...  
}
```

Recall the equals method in the Object class.
Are we overriding it here?

The Clock class has TWO equals methods.

Correct equals method

```
public boolean equals(Object x) {  
    if ( ! (x instanceof Clock)) {  
        return false;  
    }  
    Clock c = (Clock)x;  
    [Now compare the current object with c]  
}
```


Package Visibility

- What is meant by “visibility”?
- What is “package” visibility?

- Syntax:

```
public int x;  
private int y;  
int z;
```