

The background of the slide is a reproduction of a painting, likely 'Bedroom at Arles' by Vincent van Gogh. It depicts a simple bedroom with a large wooden bed frame, a small table with a chair, and another chair in the foreground. The room has a window with a view of a landscape and a coat rack with clothes hanging on it. The painting is characterized by its vibrant colors and visible brushstrokes.

# O05 Inheritance 2

java.lang.Object  
Equality

Final classes, methods and fields

Abstract classes and methods

# Object as superclass

In Java all classes ultimately inherit from **one** root class: `java.lang.Object`. Implemented methods:

- `clone()` returns copy of object
- `equals(Object obj)` establishes equivalence
- `finalize()` called by GC before reclaiming
- `getClass()` returns runtime class of the object
- `hashCode()` returns a hash code for the object
- `toString()` returns string representation of object

# Note on Equality

- Variables for **primitive types**:
  - Use `==` for equality.
  - Have no methods (i.e. have no `equals()`).
- Variables that **reference objects**:
  - `a == b`: true iff a and b reference the **same object instance**.
    - Checking the variable's immediate value is the same, which is a reference.
    - Two different instances can have exactly the same fields, and yet not be `==`.
  - `a.equals(b)`: class-specific (semantic) object equality.
    - Default inherited from `java.lang.Object` is just `==`.

# Final Classes and Methods

The `final` keyword in a class declaration states that the class **cannot** be subclassed.

The `final` keyword in a method declaration states that the method **cannot** be overridden.

# Abstract Classes and Methods

The `abstract` keyword in a class declaration states that the class is abstract, and therefore cannot be instantiated (its subclasses may be, if they are not abstract).

The `abstract` keyword in a method declaration states that the method declaration is abstract; the implementation must be provided by a subclass (like abstract methods in an interface, but here we need to be explicit and use the keyword).