Inheritance

toString() and equals()

Class Object

Single inheritance in Java, Object is the root

- Object is the root of the class hierarchy
- Every class has Object as a superclass
- All classes inherit the methods of Object but may override them
- Some methods of Object

Override equals and toString in all your classes!

Method	Behavior
`boolean equals(Object obj)`	Compares this object to its argument.
`int hashCode()`	Returns an integer hash code value for this object.
`String toString()`	Returns a string that textually represents the object.
`Class<?> getClass()`	Returns a unique object that identifies the class of this object.

An example with Pencil

- reinforce the difference between == and equals()
- Same font, same size, same color… does sharpened or usage count matter?

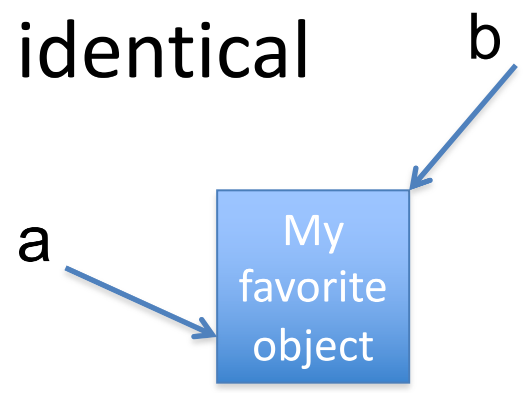
Polymorphism: Method Object.equals

- Object.equals method has a parameter of type Object

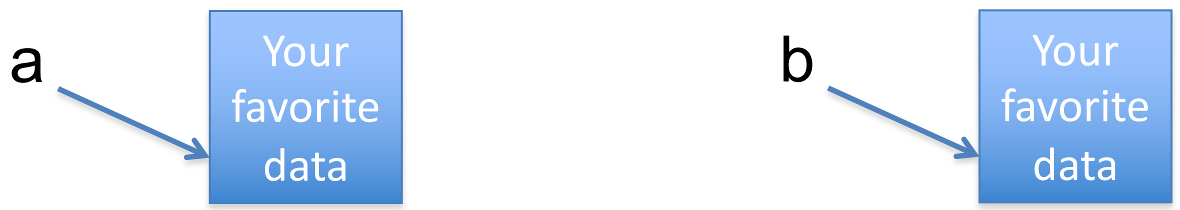
  public boolean equals (Object other) { ... }
- Compares two objects to determine if they are equal
- A class must override equals in order to support checking for equality
- Difference between identity and equality?

Identity v Equality

- a and b are identical



- And and b are equal



Equality and Identity

- Two objects are identical if they are the same object
- Two objects are equal if they "contain the same values"

```
Computer a = new Computer (…);
Computer b = a;
```

- Above, a and b are identical (only one object was allocated), that is… a and b point to the same object

If identical, then they are also equal

- **identity** ==
- **equality** a.equals(b)

Employee.equals()

```
/** Determines whether the current object matches its argument.
    @param obj The object to be compared to the current object
    @return true if the objects have the same name and address;
            otherwise, return false
*/
@Override
public boolean equals(Object obj) {
    if (obj == this) return true;
 ① if (obj == null) return false;
 ② if (this.getClass() == obj.getClass()) {
 ③      Employee other = (Employee) obj;

        return name.equals(other.name) &&
               address.equals(other.address);
 ④  } else {
        return false;
    }
}
```

Why check class?

Look, we are using equals(), which one?

Method toString

- You should always override toString method
- This allows you to print the state of the object
- This is often(unexpectedly) useful for debugging
- If you do not override it:
  - Object.toString will return a String
  - Just not the String you want!

    Example: ArrayBasedPD@ef08879

    The name of the class, @, instance’s hash code

Show equals and toString in Hokie class