

Methods (toString, equals)

CS163 Fall 2018

The toString() method

- tells Java how to convert an object into a `String`
- called when an object is printed or concatenated to a `String`:


```
Point p = new Point(7, 2);
System.out.println("p: " + p);
```
- Same as:


```
System.out.println("p: " + p.toString());
```
- Every class has a `toString()`, even if it isn't in your code.
 - The default is the class's name and a hex (base-16) hash-code:


```
Point@9e8c34
```

2

toString() implementation

```
public String toString() {
    code that returns a suitable String;
}
```

- Example: `toString()` method for our `Student` class:

```
public String toString() {
    return "name: " + name+ "\n"
        + "id: " + id + "\n"
        + "average: " + average;
}
```

- // SHOW Eclipse example of `Student` class

3

toString in ArrayLists and other collections call toString automatically

- `ArrayList<Student> students = new ArrayList<>();`
- ...
- `System.out.println(students);`
- `println(students)` calls `students.toString()`, which automatically calls `s.toString()` for every point `s`

// SHOW Eclipse example of `Student` class

4

Primitive Equality

- Suppose we have two integers `i` and `j`
- How does the statement `i==j` behave?
- `i==j` if `i` and `j` contain the same value

5

Object Equality

- Suppose we have two pet instances `pet1` and `pet2`
- How does the statement `pet1==pet2` behave?

6

Object Equality

- Suppose we have two pet instances `pet1` and `pet2`
- How does the statement `pet1==pet2` behave?
- `pet1==pet2` is true if **both** refer to the **same** object
- The `==` operator checks if the **addresses** of the two objects are equal
- May not be what we want!

7

Object Equality - extended

- If you want a different notion of equality define your own `.equals()` method.
- Use `pet1.equals(pet2)` instead of `pet1==pet2`
- The default definition of `.equals()` is the value of `==`
but for Strings the contents are compared

8

`.equals` for the Pet class

```
public boolean equals (Object other) {
    if (!other instanceof Pet) {
        return false;
    }
    Pet otherPet = (Pet) other;
    return ((this.age == otherPet.age)
        &&(Math.abs(this.weight - otherPet.weight) < 1e-8)
        &&(this.name.equals(otherPet.name)));
}
```

// SHOW ECLIPSE EXAMPLE OF Equals code.

9