CS 302 Week 9

Jim Williams
This Week

P2 Milestone 3
Lab: Instantiating Classes
Lecture:
  ● Wrapper Classes
  ● More Objects (Instances) and Classes

Next Week: Spring Break
Will this work?

Double d = new Double(10);
double d2 = d;

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>yes</td>
<td></td>
</tr>
<tr>
<td>no</td>
<td></td>
</tr>
<tr>
<td>sometimes</td>
<td></td>
</tr>
<tr>
<td>error</td>
<td></td>
</tr>
</tbody>
</table>
### Is result true or false?

Integer m = 5;
Integer n = 5;
boolean result = m == n;

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td></td>
</tr>
<tr>
<td>false</td>
<td></td>
</tr>
<tr>
<td>error</td>
<td></td>
</tr>
</tbody>
</table>

Classes and Objects

various variable declarations
accessors (getters), mutators (setters)
constructor, no-arg, default
public, private, protected, <package>
state of an object
constructor overloading
this
Memory areas

code

static

heap

stack
class Pizza {
    private String toppings;
    public String getToppings() {
        if (toppings == null) {
            toppings = "no toppings";
            return toppings;
        }
        return toppings;
    }
}
The visibility modifier **private** is appropriate.

```java
class Pizza {
    private String toppings;
    public String getToppings() {
        if (toppings == null) {
            toppings = "no toppings";
        }
        return toppings;
    }
}
```

This doesn't allow a user of this class to change the field (attribute) directly, but allows only methods within this class to change the field.
A static field means

class Pizza {
    static String toppings;
    public String getToppings() {
        if (toppings == null) {
            toppings = "no toppings";
        }
        return toppings;
    }
}

toppings is a class variable

can be accessed by any method

cannot be changed

every instance will have the exact same toppings
getToppings() is a(n) method

class Pizza {
    private String toppings;
    public String getToppings() {
        if (toppings == null) {
            toppings = "no toppings";
            return toppings;
        }
        return toppings;
    }
}
getToppings()

class Pizza {
    private String toppings;
    public static String getToppings() {
        if (toppings == null)
            toppings = "no toppings";
        return toppings;
    }
}
State of an instance

The current values of its attributes.
Changing the values changes the state.
Ball Example

class Ball {
    boolean inflated;
}

//main

    Ball b = new Ball();
    b.inflated = true;
    Ball b2 = new Ball();
setToppings

class Pizza {
    private String toppings;
    public void setToppings( String tops) {
        if ( tops != null && tops.length() > 0) {
            toppings = tops;
        }
    }
}
this

The way to access the current instance from within the instance methods.
constructor

Called when instance created
Used to initialize instance fields
Can a setter also get?

class Pizza {
    private String toppings;
    public String setToppings( String tops) {
        String previous = toppings;
        toppings = tops;
        return previous;
    }
}

<table>
<thead>
<tr>
<th></th>
<th>yes</th>
<th>no</th>
<th>depends</th>
<th>error</th>
</tr>
</thead>
</table>

enum Color {GREEN, YELLOW, RED};
class TrafficLight{
    private Color color = Color.RED;
    public void change() {
        if (color == Color.GREEN) color = Color.YELLOW;
        else if (color == Color.YELLOW) color = Color.RED;
        else color = Color.GREEN;
    }
    public Color getColor() { return color; }
    public String toString() { return color.name(); }
}
What variables are used for state in "instance who's reference is in p"?

class Person {
    private static int count = 0;
    private boolean amHungry = false;
    public Person( String name) {
    }
}

public static void main(String []args) {
    Person p = new Person("Fred");
}
Visibility Modifiers

For members of a class:

- public
- private
- protected
- <package>

Demo
Can methodA call methodB?

//classes in different files in same package
class A {
    public void methodA() {
        B b = new B();
        b.methodB();
    }
}
class B {
    void methodB() {
    }
}
Can a method outside the package call methodA()?

//classes in different files in same package

class A {
    public void methodA() {
        B b = new B();
        b.methodB();
    }
}

class B {
    void methodB() {
    }
}
Bike

numWheels

numGears
Will toString return results of Graphic toString()?

class Bird {
    private Graphic graphic;
    Bird(String name) {
        this.graphic = new Graphic(name);
    }
    public String toString() {
        return this.graphic.toString();
    }
}

yes
no
NullPointerException
other
class Picture {
    private boolean hasFrame;
    public Picture(boolean hasFrame) {
        this.hasFrame = hasFrame;
    }
}
Does this print true or false?

class Person {
    static int count = 0;
    private boolean something = false;
    boolean getThing(boolean something) {
        return this.something;
    }
}

Person p = new Person();
System.out.println( p.getThing( true));
Does this print 0, 1, other or error?

class Person {
    static int count = 0;
    private boolean something = false;
    Person(boolean something) {
        this.something = something;
        count++;
    }
}
System.out.println(Person.count);