Lec 9

Inheritance
• https://www.youtube.com/watch?v=QkkoHAnjUs
• http://i.imgur.com/OZtqOxq.gifv
• http://i.imgur.com/lvmN4Cw.jpg
• http://bit.ly/1KBnVJC
Car

- topSpeed: int
- bodyType: String
- color: String
- damage: int

+ Car()
+ Car(topSpeed: int, bodyType: String, color: String, damage: int)
+ accelerate()
+ reverse()
+ brake()
Tank

- topSpeed:int
- bodyType:String
- color:String
- damage:int
- ammunition:int

+ Tank()
+ Tank(topSpeed:int, bodyType:String, color:String, damage:int)
+ accelerate()
+ reverse()
+ brake()
+ Fire()
public class Tank extends Car
class RX7 extends Car {
    private int NOS;
    public RX7() {
        NOS = 10;
    }

    public void useNOS() {
        System.out.println("hold on");
        NOS--;
        accelerate()
    }
}

class Car {
    private String color;
    public Car() {
        color = "RED";
    }
    public void accelerate() {
        System.out.println("let’s GO");
    }
}

class Start {
    public static void main(...) {
        RX7 dom = new RX7();
        dom.useNOS();
        dom.accelerate();
    }
}
Inheritance

https://www.youtube.com/watch?v=KgJs95dzFE0
class RX7 extends Car {
    private int NOS;
    public RX7() {
        NOS = 10;
    }

    public void useNOS() {
        System.out.println("Hold on");
        NOS--;
        accelerate()
    }
}

class Car {
    private String color;
    public Car() {
        color = "RED";
    }

    private void accelerate() {
        System.out.println("Let’s GO");
    }
}

class Start {
    public static void main(...) {
        RX7 dom = new RX7();
        dom.useNOS();
        dom.accelerate();
    }
}
Inheritance – Key Points

Superclass

Subclass

class Superclass

class Subclass extends Superclass

Subclass **is-a** Superclass

Not the other way around

Superclass members become members of the Subclass (inheritance)

• except private methods

• private instance variables accessible through public superclass methods

Implicit **extends** Object if no explicit extends
Details

• Can only extend one class
  – But that class could extend it’s own class, e.g.:

```java
public class USTank extends Tank {}
public class Tank extends Vehicle {}
public class Vehicle {}
```

• Eventually, every class extends Object
  – if not explicitly, then implicitly
this vs. super

this – refer to this instance, e.g. this.color = color

super – refer to parent’s (aka super’s) fields/method

this() – same class constructor invocation

super() – immediate superclass constructor invocation
this vs. super

Since most instance vars are private, we use `super(args);` to pass values to initialize our super class's private instance vars.

Use `this.` to explicitly specify instance members vs. local vars or parameters.

```java
public class Car {
    private String color;

    public Car(String color) {
        this.color = color;
    }

    public int getColor() { return color; }
}
```

```java
public class Tank extends Car {
    private int ammo;

    public Tank(String color, int ammo) {
        super(color);
        this.ammo = ammo;
    }
}
```

```java
Tank tank = new Tank(“Green”, 10);
```
What is the value of the instance variable `color` if Superclass's ctor was written as below?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>green</td>
</tr>
<tr>
<td>B</td>
<td>blue</td>
</tr>
<tr>
<td>C</td>
<td>null</td>
</tr>
<tr>
<td>D</td>
<td>Compile error</td>
</tr>
<tr>
<td>E</td>
<td>Runtime error</td>
</tr>
</tbody>
</table>

```java
public class Car {
    private String color = "blue";

    public Car(String color) {
        color = color;
    }

    public String getColor() { return color; }
}

public class Tank extends Car {
    private int ammo;

    public Subclass(String color, int ammo) {
        super(color);
        this.ammo = ammo;
    }
}
```

```java
tank = new Tank( "green", 10 );
System.out.println(tank.getColor());
```
this vs. super

What is the value of the instance variable color if Superclass's ctor was written as below

A) green
B) blue
C) null
D) Compile error
E) Runtime error

```java
public class Car {
    private String color = "blue";

    public Car ( String color ) {
        this.color = color;
    }

    public String getColor() { return color; }
}

public class Tank extends Car{
    private int ammo;

    public Tank( String color, int ammo) {
        super( this.color);
        this.ammo = ammo;
    }

    Tank tank = new Tank( "green", 10 );
    System.out.println(tank.getColor());
```
this vs. super

What is the value of the instance variable \textit{color} if Superclass's ctor was written as below

A) green
B) blue
C) null
D) Compile error
E) Runtime error

```java
public class Car {
    private String color = "blue";
    public Car ( String color ) {
        this.color = color;
    }
}

public class Tank extends Car{
    private int ammo;
    public Subclass( String color, int ammo) { 
        super( color);
        this.ammo = ammo;
    }
    public String getColor() { return color ; } 
}
```

```java
Tank tank = new Tank("green", 10 );
System.out.println(tank.getColor());
```