

CSSE 220 – Object-Oriented Software Development
 Rose-Hulman Institute of Technology

Worksheet 15

Name (Print): _____ Section: _____

```
interface Top {
    public void alpha();
    public void beta();
    public void gamma();
    public void delta();
    // Note no epsilon here
}
```

```
class One implements Top {

    public void alpha() {
        System.out.println("A");
    }

    public void beta() {
        System.out.println("B");
    }

    public void gamma() {
        System.out.println("C");
    }

    public void delta() {
        System.out.println("D");
        this.beta();
    }
}
```

```
class Two extends One
    implements Top {

    public void beta() {
        System.out.println("E");
    }

    public void gamma() {
        super.gamma();
        System.out.println("F");
    }

    public void epsilon() {
        System.out.println("G");
    }
}
```

1.

```
Two m = new Two();
Top q = new One();
Top r = new Two();
One s = new Two();
```

<code>((Two) r).epsilon();</code>	<i>no</i>	<i>runtime</i>	<i>compile</i>
A B C D E F G BE CF DB DE EB FC	<i>output</i>	<i>error</i>	<i>error</i>
<code>((Two) q).epsilon();</code>	<i>no</i>	<i>runtime</i>	<i>compile</i>
A B C D E F G BE CF DB DE EB FC	<i>output</i>	<i>error</i>	<i>error</i>
<code>Two w = new One();</code>	<i>no</i>	<i>runtime</i>	<i>compile</i>
A B C D E F G BE CF DB DE EB FC	<i>output</i>	<i>error</i>	<i>error</i>
<code>One x = new Two();</code>	<i>no</i>	<i>runtime</i>	<i>compile</i>
A B C D E F G BE CF DB DE EB FC	<i>output</i>	<i>error</i>	<i>error</i>
<code>Top y = new Top();</code>	<i>no</i>	<i>runtime</i>	<i>compile</i>
A B C D E F G BE CF DB DE EB FC	<i>output</i>	<i>error</i>	<i>error</i>
<code>r.delta();</code>	<i>no</i>	<i>runtime</i>	<i>compile</i>
A B C D E F G BE CF DB DE EB FC	<i>output</i>	<i>error</i>	<i>error</i>

<pre> public interface Performer { String rehearse(); void perform(); } class Person { public String rehearse() { return "Just Sit."; } } class Dancer extends Person implements Performer { public String rehearse() { return "Dance!"; } public void perform() { System.out.print(rehearse()); } } </pre>	<pre> class Actor extends Dancer { public String rehearse() { return "Sing and " + super.rehearse(); } } class Star extends Actor { private Dancer backupDancer; Star(Dancer b){ this.backupDancer = b; } public void perform() { System.out.print(rehearse()); this.backupDancer.perform(); } } </pre>
---	---

2.

```

Person a = new Actor();
Dancer d = new Dancer();
Performer s = new Star(d);

```

a.rehearse(); _____ *no runtime compile*
output error error

a.perform(); _____ *no runtime compile*
output error error

(Dancer)a.perform(); _____ *no runtime compile*
output error error

((Star)a).perform(); _____ *no runtime compile*
output error error

d.perform(); _____ *no runtime compile*
output error error

s.perform(); _____ *no runtime compile*
output error error

((Person)s).perform(); _____ *no runtime compile/*
output error error