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

Worksheet 18

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();

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

2.

```
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()  );
    }
}
```

```
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();
    }
}
```

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

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

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

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

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

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

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

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