Type checking exercises

1. What does the `main()` function of the following code print? Why?

```
class Plant {
    String leaf() { return "plant leaf\n"; }
    String leaf(Plant other) {
        return "plant leaf with other: " + other.leaf();
    }
    public static void main(String[] args) {
        Plant p = new Rose();
        Plant p2 = new Tulip();

        System.out.println( p.leaf(p2) );
        System.out.println( p2.leaf(p) );

        // Notice that this cast goes "down" the inheritance
        // hierarchy, and so is legal; however, it would raise
        // ClassCastException at runtime if p2 did not point to an
        // actual Tulip.
        Tulip t = (Tulip)p2;
        System.out.println( t.leaf(new Rose()) );
    }
}

abstract class Flower extends Plant {
    abstract String leaf(Flower other);
    String leaf() { return "flower leaf\n"; }
    String leaf(Plant other) {
        return this.leaf(this)
               + "flower leaf with other: " + other.leaf();
    }
}

class Tulip extends Flower {
    String leaf() { return "tulip leaf\n"; }
    String leaf(Flower other) {
         return "tulip leaf with other: " + other.leaf();
    }
    String leaf(Rose other) {
        return "tulip leaf with rose other\n";
    }
}

class Rose extends Flower {
    String leaf(Flower other) {
         return "rose leaf, " + this.leaf()
                + "with other: " + other.leaf();
    }
    String leaf(Tulip other) {
         return "rose leaf with tulip other: "
                + other.leaf(this);
    }
}
```

2. Given the hierarchy of classes in the previous question, which of the following lines raises a static type error? Which will raise a runtime class cast exception? Which method will be called for legal methods?

```
Tulip t = new Tulip();
Plant p = t;
Flower f = p;
Flower f2 = (Flower)p;

Rose r = (Rose)p;
Rose r2 = new Rose();
r.leaf(p);
r.leaf(r2);

Flower f3 = r;
Flower f4 = (Tulip)p;
f4.leaf( (Plant)f3 );
f4.leaf( (Rose)f3 );
f4.leaf( (Rose)(Tulip)f3 );
```

3. Explain the bug in each of the following two classes (for the second class, it is not a type bug):

```
class Orchid extends Flower {
    String leaf(Plant other) {
        return "orchid leaf with plant other: " + other.leaf();
    }
}

class Violet extends Flower {
    String leaf() { return this.leaf((Flower)this); }
    String leaf(Flower other) {
        return this.leaf((Plant)other);
    }

    public static void main(String[] args) {
        Violet v = new Violet();
        Plant p = new Plant();
        v.leaf(p);
    }
}
```