

CSE 143 Polymorphism Handout

Consider the following classes:

```
public class Snow {
    public void method2() {
        System.out.println("Snow 2");
    }

    public void method3() {
        System.out.println("Snow 3");
    }
}

public class Rain extends Snow {
    public void method1() {
        System.out.println("Rain 1");
    }

    public void method2() {
        System.out.println("Rain 2");
    }
}

public class Sleet extends Snow {
    public void method2() {
        System.out.println("Sleet 2");
        super.method2();
        method3();
    }

    public void method3() {
        System.out.println("Sleet 3");
    }
}

public class Fog extends Sleet {
    public void method1() {
        System.out.println("Fog 1");
    }

    public void method3() {
        System.out.println("Fog 3");
    }
}
```

Suppose the following variables are defined:

```
Snow var1 = new Sleet();
Snow var2 = new Rain();
Sleet var3 = new Fog();
```

In the table below, indicate in the right-hand column the output produced by the statement in the left-hand column. If the statement produces more than one line of output, indicate the line breaks with slashes as in "a / b / c" to indicate three lines of output with "a" followed by "b" followed by "c". If the statement causes an error, fill in the right-hand column with the phrase "ERROR" to indicate this.

Statement

- a) var1.method2();
- b) var2.method1();
- c) var2.method2();
- d) var3.method2();
- e) var3.method3();
- f) ((Rain) var2).method1();
- g) ((Sleet) var1).method3();
- h) ((Sleet) var1).method1();
- i) ((Sleet) var2).method3();