1. [40%] The interface TempI is to convert between degrees Fahrenheit (F) and Centigrade (C). Recall $f = \frac{9}{5}c + 32$ and $c = \frac{5}{9}(f - 32)$.

```java
public interface TempI {
    public void setFahrenheit(); // to convert C to F
    public void setCentigrade(); // to convert F to C
    public float convert(float v); // does conversion
    public boolean freezing(float v); // true if v is freezing or below
}
```

usage:
```
TempI t = new Temperature(); // Temperature implements TempI
float v = 20.0f; // will be regarded as C then F

    t.setCentigrade(); // first interpret arguments as C.
    t.freezing(v);     ==> false // so 20 deg C. is above freezing
    t.convert(v);      ==> 68.0 // and 20 deg C. = 68 deg F.
    t.setFahrenheit(); // now interpret arguments as F.
    t.freezing(v);     ==> true; // 20 deg F. is freezing
    t.convert(v);      ==> -6.7; // and 20 F. = -6.7 C.
```

1a [15%] Write the class Temperature using the pure State Pattern (do not use instanceof, and use conditionals only to make numeric comparisons).
public class Temperature implements TempI {

    public Temperature() {

    }

    public void setFahrenheit() {

    }

    public void setCentigrade() {

    }

    public float convert(float v) {

    }

    public boolean freezing(float v) {

    }
lb. [25%] Write **State**, **CentigradeState**, and **FahrenheitState** below:
2.[60%] **Sunshine** and **Rain** are two kinds of **Visitor** to **Cat** and **Rose**. For a Cat, Sunshine makes it purr and Rain makes it hiss. A Rose blooms after both Sunshine and Rain have visited (isblooming returns true).

2a [10%] First have rain and sun visit both cat and rose:

```java
Cat cat = new Cat();
Rose rose = new Rose();
Visitor sun = new Sunshine();
Visitor rain = new Rain();
```

2b [20%] Next, complete **Cat** and **Rose** according to the **Visitor Pattern**:

```java
public class Cat {

    public Cat()

    public void purr() { System.err.println("purr"); }
    public void hiss() { System.err.println("hiss"); }

}

public class Rose extends Plant {
    private boolean gotRain = false;
    private boolean gotSun = false;

    public Rose(

    public void receiveRain() { gotRain = true; }
    public void receiveSunshine() { gotSun = true; }
    public boolean isBlooming() { return gotRain && gotSun; }

}
```
2b [30%] Now write: a) **Visitor**, b) **Sunshine** and c) **Rain** and assume that they only visit Cat and Rose. Make sure, as mentioned: Sunshine makes a Cat purr() and Rain makes it hiss(). For a Rose, after **both** Sunshine and Rain have visited, the Rose will bloom (isblooming() would return true).