Meet the Template Method

We've basically just implemented the Template Method Pattern. What's that? Let's look at the structure of the CaffeineBeverage class; it contains the actual "template method:"

```java
public abstract class CaffeineBeverage {

    void final prepareRecipe() {
        boilWater();
        brew();
        pourInCup();
        addCondiments();
    }

    abstract void brew();
    abstract void addCondiments();
    void boilWater() {
        // implementation
    }
    void pourInCup() {
        // implementation
    }
}
```

prepareRecipe() is our template method. Why?

Because:

1. It is a method, after all.
2. It serves as a template for an algorithm, in this case, an algorithm for making caffeinated beverages.

In the template, each step of the algorithm is represented by a method.

Some methods are handled by this class...

...and some are handled by the subclass.

The methods that need to be supplied by a subclass are declared abstract.

**The Template Method defines the steps of an algorithm and allows subclasses to provide the implementation for one or more steps.**