Abstract Classes and Interfaces Assignment

**DinnerTable Assignment**

You are to create a project that will have 7 classes and 1 interface.

The seven classes are shown below, with their inheritance structure:

EatingUtensil Fork SaladFork

DinnerFork

Spoon SoupSpoon

Knife

EatingUtensil is an abstract class that will have 3 data fields, name (String), *dirty* (boolean) and *location* (int). The EatingUtensil class will also have two methods: *setName*, and *place*, which places the EatingUtensil on the DinnerTable. The *place* method receives an integer value, from 1 to 5, based on its order from left to right and assigns it to *location*.

1 2 3 4 5

Fork will also be an abstract class and will not be instantiated. It should have one data field (beyond those inherited): *numTines*, which specifies the number of tines on the fork. For our program we will assume that DinnerForks have five tines and SaladForks have four tines.

SaladForks and DinnerForks will have the ability to be placed, washed and used (eaten with).

Spoon is not abstract and represent a typical spoon. SoupSpoon should inherit Spoon and add a *declineSoup* method that will remove the SoupSpoon from the table. Like the Forks, both spoon types should have the ability to be placed, washed and eaten with.

The *Knife* is just a knife. It can be placed, washed or eaten with also. Additionally, create a *cut* method that will output "You cut up your food." to the console.

The interface that needs to be implemented is *Eating*. This interface should implement the *wash* and *eatWith* methods. All classes that can be instantiated (non-abstract) should implement this interface.

Create 2 runners that two different place settings. Each runner should test and output the various data fields of the EatingUtensils in your project.