# List of Contract Deviations

1. Removed the abstract classes from our design. Ruby does not have any notion of interfaces, or abstract classes due to its "Duck Typing" feature, and we want our factory and solver classes to be able to support Duck Typing as well.

2. Removed all factory classes from our design. Factory classes made sense in other object oriented languages where they needed to adhere to an interface, so that the compiler knows that the classes are capable of what they set out to do. Ruby does not use interfaces, and as a result, each class's initialization method is it's own factory method.

3. Removed all contractual checks that use `.kind_of?` method. Like change #1, we do not want to limit the use of our library only on types we define.

# Additional Testing Beyond Contracts

In order to aid our programming correctness, we have created test cases for all our test classes. We aim to achieve 100% code coverage with our tests.

Our tests are set up to use the "SimpleCov" gem, which provides HTML output showing our coverage statistics. Our tests are linked together in the `master_test_suite.rb` file, which includes a reference to all our testing code.