QuantFinanceFramework (QFF) Development Guide

# Style

## Test-driven development (TDD)

Test-driven development starts with developing test for each one of the features. The test might fail as the tests are developed even before the development. Development team then develops and refactors the code to pass the test.

Test-Driven Development Process:

1. Add a Test
2. Run all tests and see if the new one fails
3. Write some code
4. Run tests and Refactor code
5. Repeat

# Programming Principles

## DRY principle

Don't repeat yourself

## S.O.L.I.D principles

* S — Single responsibility principle

A class should have one and only one reason to change, meaning that a class should only have one job.

* O — Open closed principle

Objects or entities should be open for extension, but closed for modification.

* L — Liskov substitution principle

every subclass/derived class should be substitutable for their base/parent class.

* I — Interface segregation principle

A client should never be forced to implement an interface that it doesn’t use or clients shouldn’t be forced to depend on methods they do not use.

* D — Dependency Inversion principle

Entities must depend on abstractions not on concretions. It states that the high-level module must not depend on the low-level module, but they should depend on abstractions.

## KISS principle

Keep it simple, stupid! Premature optimization is the root of all evil!

# Programming Rules

* Each class can be no longer than a hundred lines of code.
* Each method can be no longer than five lines of code.
* No more than four parameters for each method.
* You can break the rules if you can explain the benefit of doing so to the team.