Quality Assurance Agreement

This document outlines what we have decided as a team so that the code we create is quality and so everyone in the team is able to understand the code that is written.

1. Pair Programming : allows for instance code review.
2. Refactoring: to improve the code quality and overall design. We decided that this should be done by someone other than the person who coded in the first place. This allows for a different person to offer a different perspective to how that particular feature should be coded.
3. Involve the End-User/User Testing: If there is any doubt over how a feature should be designed or the understanding of the feature we must first consult the user.
4. Naming Conventions:

* avoid abbreviations where possible. Only use know abbreviations such as itr for iterator.
* variable, class, and method names should convey the purpose of what it is and also so it is readable and understandable.
* classes: Car {}
* variables: maxCount;
* methods: getCount();

1. Unit Testing: code should go through a automated test to check the behaviour of that code is what is expected and correct.
2. Test-Driven Development: write tests which fail, produce the minimum amount of code to pass the test, and finally refactor the new code to the appropriate standards.
3. Continuous Integration: making sure code builds and running the programming so it works and doesn't break any existing code.