What is defensive coding?

Making software behave in a predictable manner despite unexpected inputs or user actions. Making the source code readable and understandable. Reducing the number of software bugs and problems.

Quality can be improved through automated code testing.

Clean Code – improves comprehension, simplifies maintenance and reduces bugs.

Unit testing – Improves quality, confirms maintenance and reduces bugs.

Validation & Exception handling – Improves predictability, makes code more consistent and reduces bugs

Clean code :

* write good code using clean coding techniques.
* Transform bad code into good code with refactoring
* Easy to read. Developers can understand and not introduce bugs over time. It’s more tolerant to quick changes
* Clear intent. It’s structure and simple names add to the understanding. If the code is too complex or too elegant to understand, it’s harder to maintain.
* Simple
* Minimal. Each part of the code does one thing well and dependencies are kept to a minimum
* Thoughtful. Code location is considered. Code divided into logical classes and methods

Testable code and unit tests:

* Makes testing simple. Only need to test the part of the code you’re working on.

Validation & exception handling:

* Making predictable code
* Should the code trust the data? What errors could occur?

Each non-private method written is a contract. A promise made to other parts of the code. It promises to accept a set of params with defined data types and it promises to return a type or no value and it promises to only throw a specific set of defined and anticipated exceptions.

The code should trust but verify.

* The params
* State of invariance. Global or class level data.
* Return predictable results
* Throw expected exceptions.

Clean yet protected?

We must minimise the amount of code but have enough to deal with different eventualities. You want the code to be protected but not so protected that its sluggish and hard to work with.

The more lines of code, the more chance for error.