**Chapter-1 Clean Code**

**There Will Be Code**

* Code represent the detail of requirements some level detail cannot ignored when it is executing that is programming
* Abstraction(quality) of our language will increase ,domain-specific languages will grow,but it will not eliminate the code
* Code should be in a formal
* Code really language which ultimately express the requirement
* never eliminate necessary code - so there will be code

**Bad Code**

* In the late 80s (killer app) is shut down because of bad code
* huge mess in code when added more feature code got worse , this could not manage so long - bad code will bought your product down
* obstraction many times that is **wadding** that time you go throught the bad code -strugle to find hoping some hints and clues. we will see many sensless code
* Dont leave the messy code to be cleaned later , Later equals never

**Total Cost of Owning a Mess**

* Messy code be very fast at beginning of a project but moving in a snail pace
* Every change in the code add the knot in code
* mess of code make productivity will be zero , management will add the more staff to the project that make more difficult for the all so that will not be solution
* **The Grand Redesign in the sky**

Redesign takes more time rather we can clean that code that could be better

Spending time keeping your code clean is not just cost efficient it matters of professional survival

* **Attitude**
* Never blame others for your messy code , comminucate your doubts without shyness
* In product everyone should clean code because every one wants the clean code
* We should always explain about the messy of code to the manager it is matter of proffesionalism
* **The Primal Conundrum**
* The way to make the deadline to finish only one way to go fast is to keep the code as clean at all time
* **The Art of Clean code**
* Only way go fast keep your code clean
* Writing clean code is lot like painting a picture
* Code sense is the key ,some born with it,some fight to acquire it
* Who writies the clean code is an artist
* **What is clean code**
* **Bjarne Stroustrup**
* Bad code temps to mess code grows
* Clean code should simple and efficient
* error handling should be completed
* discpline of paying attention to the details
* should as minimal dependency
* **Grady Booch**
* Clean code is simple and direct
* Clean code reads like well written prose(Spoken language)
* It should be crisp(without unecessary detail) abstraction(quality) and straight forard lines of control
* **“Big” Dave Thomas**
* Clean code can be read and enhanced by developer rather than original author
* Smaller code is better for clean code
* Meaningful names , code without testcase is not clean
* **Michael Feathers**
* Clean code always looks like it was written by someone who cares
* No one can enhance better
* **Ron Jeffries**
* Runs all the testcase
* contains no duplication
* Express all the design that are in the system
* minimize the number of entities such as classes methods fuction a like
* **Ward Cunningham**
* Clean code is called beautiful code
* **Schools of Thoughts**
* To make clean code - clean variable name , clean function ,clean class etc...
* we are only right everyone has their unique way of teaching and they are also right.
* **We are authors**
* next time you write a line of code remeber your an author writing for readers who will judge the efforts
* making it easy to read actually makes it easier to write
* you cannot write the code if you cannot read the surrounding code
* **The Boy Scoute Rule**
* Leave the campground cleaner than you found it - dont always
* change one variable name for better
* break up one function that is to large
* clean up small duplications
* clean up composite if statement