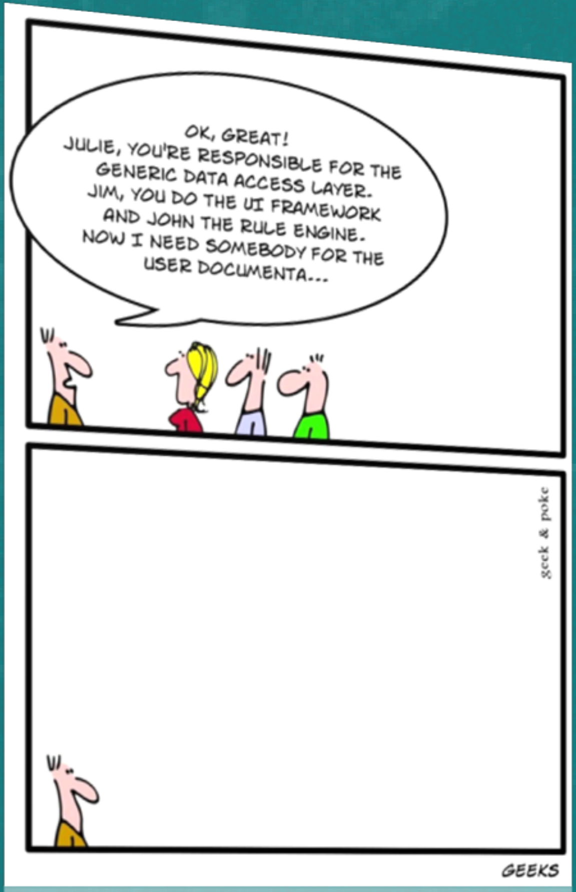

OK, GREAT!
JULIE, YOU'RE RESPONSIBLE FOR THE GENERIC DATA ACCESS LAYER.
JIM, YOU DO THE UI FRAMEWORK AND JOHN THE RULE ENGINE.
NOW I NEED SOMEBODY FOR THE USER DOCUMENTA...
geek & poke
GEEKS

好记性不如烂笔头

The Palest Ink Is Better Than The Best Memory

Project Management Tools: Documentation and Version Control

Production and Project Management

Project Management Tools

Documentation

What are Documentation?

- Project Document
  - Pitch Document
  - Project Plan/Design Document/Technical Document
- Work Plan/Schedule/PERT/Gantt
  - Estimates, Check List
- Risk Management
- Change Control
- What else?

What else?

- Daily Report
- Progress Updates
  - Check List
- Issues/Bugs Tracking
- Project Report
- Post Mortem/Closure
  - Lesson learnt
- Code comments (yes! This too!)
  - Version Control

Why we don’t document

Issues about Documentation

“Hmmm…I would prefer to use that time for project development.”

“Too much paperwork, it cause delays and rise cost?”

“We’ll do it after completing the project.”

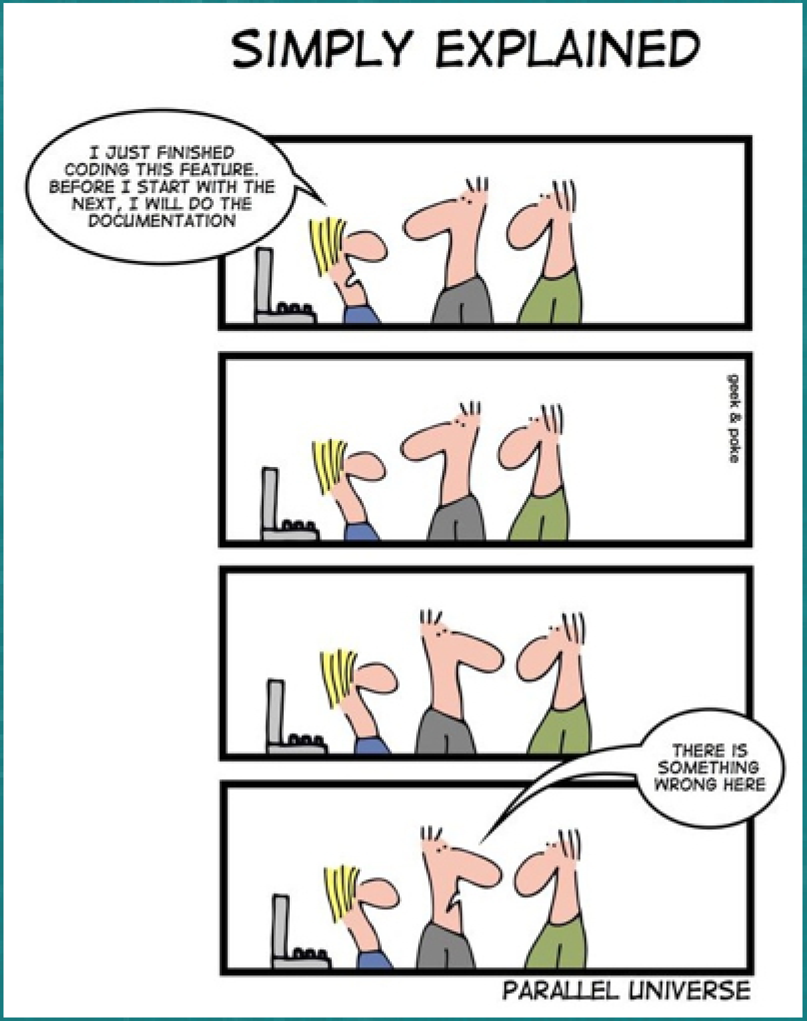
“I would like to focus on delivering the project. Documentation can wait.”

“I can complete one more project if I don’t do documentation.”

“Huh, I am writing great, clean, beautiful code; people will never have problems reading this.”

Issues about Documentation

- Too administrative
- Rise project cost due to delays
  - Since we need to spend time on it!
  - Programmers' time are expensive!
- Slows down the entire process
- Lack standards
- Poor quality
- Not available/accessible!


SIMPLY EXPLAINED
I JUST FINISHED CODING THIS FEATURE. BEFORE I START WITH THE NEXT, I WILL DO THE DOCUMENTATION
geek & poke
THERE IS SOMETHING WRONG HERE
PARALLEL UNIVERSE

Why we should document

Must I document?

I remember...

- Even if you have a great memory, you won't be able to remember everything that was said, unless you have a permanent written record for your reference.

- What animals are there in the farm?
  - Remember, don’t look at the pictures!

A story about a farm

Name the animals?

Old Macdonald own a farm. And on her farm are 2 pigs, 4 horses, 3 cats, 4 chickens, 1 duck, 3 goats and 2 dogs.

Don’t look at the picture!

I forget…

Who owns the farm?

How many cows are there in her farm?

For Yourself

- It’s NOT all about memory!
- After a month:
  - How did I solve this problem ?
  - Why does it work ?
  - Why do I need this line ?
  - What’s the input ?
  - What’s the output ?
- Protecting Yourself
  - from false accusations; your word against somebody else’s
  - Proper documentation protects you!

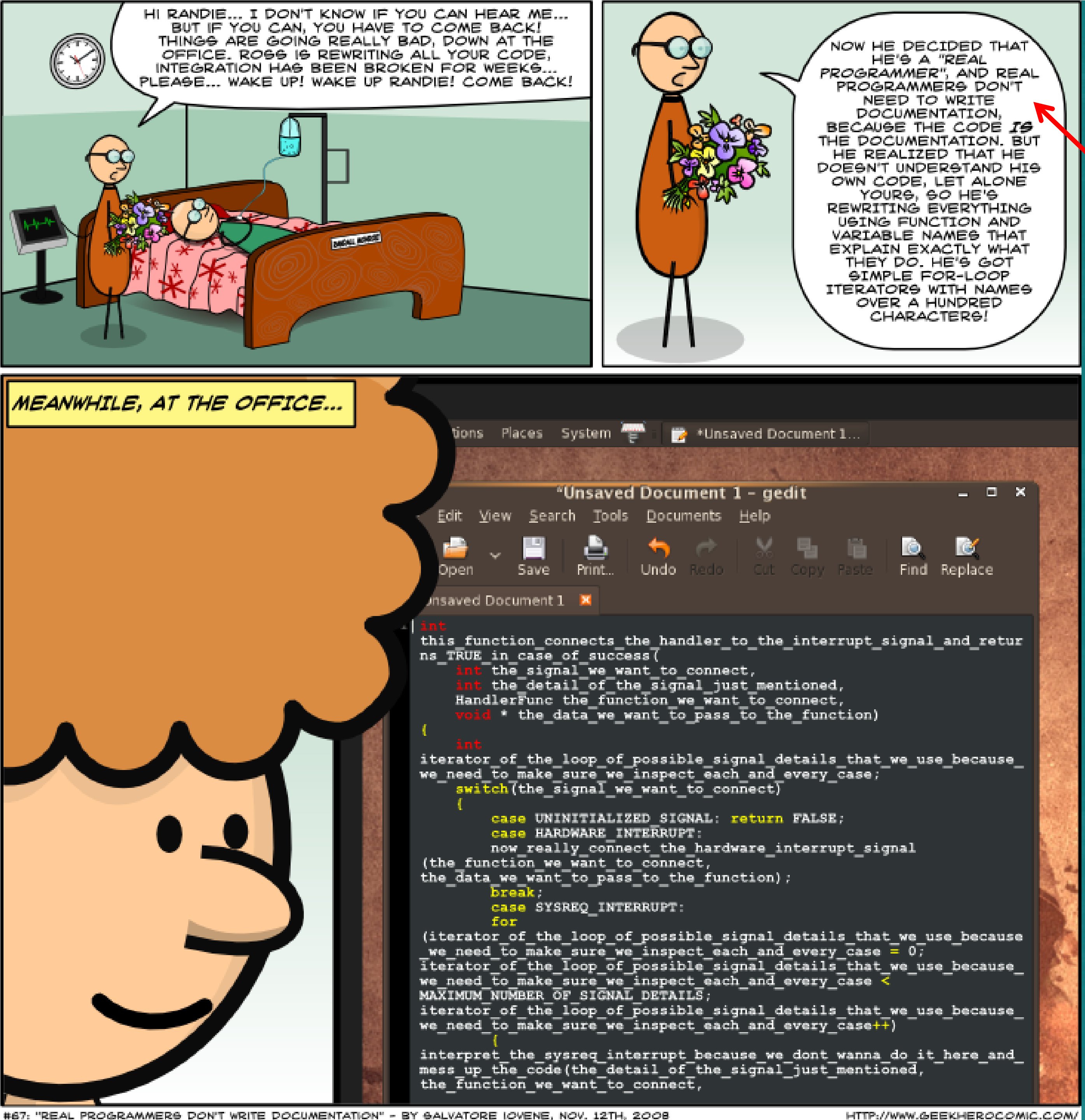
Can you remember what I said last week about assessing Risk?

For the Team

- Facilitates team development
  - Identifies what the project is and is not
  - Clarifies issues the project team may have
- Pivotal communication medium
  - Recorded Information reaches the members are timely, accurate and complete
- Measure progress toward completion
  - keep track of changes
- Provides traceability of what was said
  - Eliminate faulty human memories

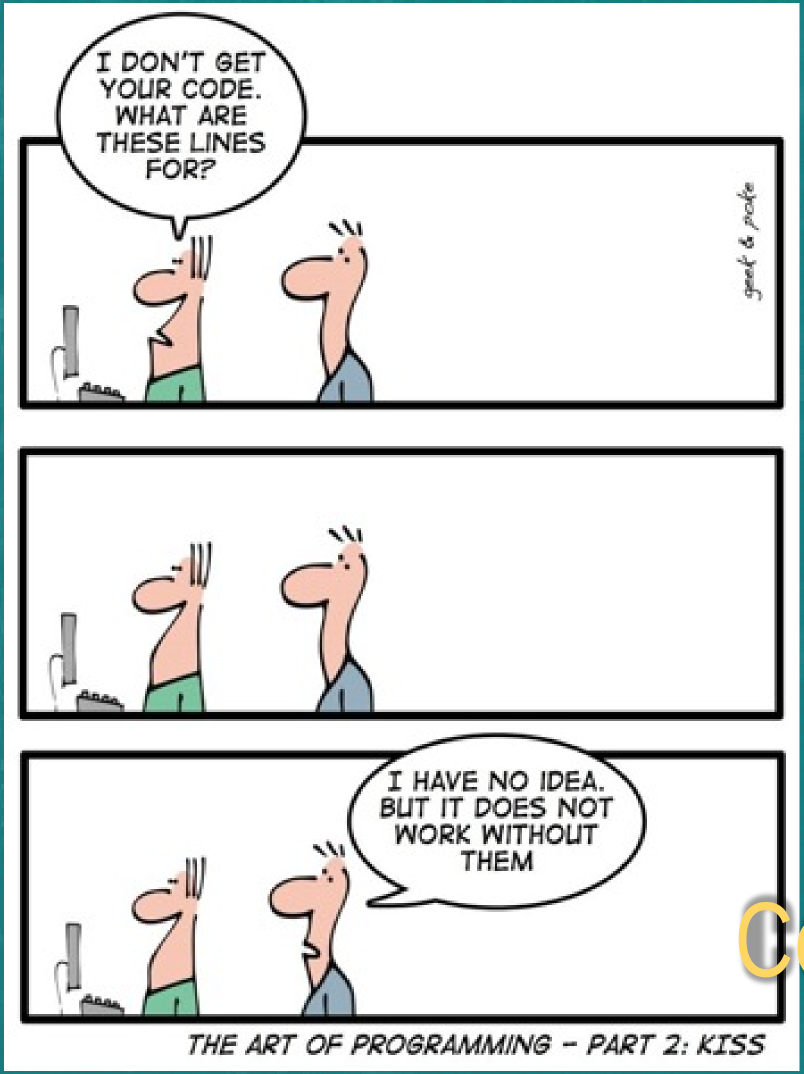
For Better Management

- To provide clarity and traceability
  - Keep stakeholders and other members know your work/update on progress
- Maybe you won’t be around
  - On holiday?
- Continuity – Handling over
  - Promotion….


HI RANDIE... I DON'T KNOW IF YOU CAN HEAR ME... BUT IF YOU CAN, YOU HAVE TO COME BACK! THINGS ARE GOING REALLY BAD, DOWN AT THE OFFICE. ROSS IS REWRITING ALL YOUR CODE, INTEGRATION HAS BEEN BROKEN FOR WEEKS... PLEASE... WAKE UP! WAKE UP RANDIE! COME BACK!
NOW HE DECIDED THAT HE'S A "REAL PROGRAMMER", AND REAL PROGRAMMERS DON'T NEED TO WRITE DOCUMENTATION, BECAUSE THE CODE *IS* THE DOCUMENTATION. BUT HE REALIZED THAT HE DOESN'T UNDERSTAND HIS OWN CODE, LET ALONE YOURS, SO HE'S REWRITING EVERYTHING USING FUNCTION AND VARIABLE NAMES THAT EXPLAIN EXACTLY WHAT THEY DO. HE'S GOT SIMPLE FOR-LOOP ITERATORS WITH NAMES OVER A HUNDRED CHARACTERS!
MEANWHILE, AT THE OFFICE...
tions Places System
*Unsaved Document 1...
*Unsaved Document 1 - gedit
Edit View Search Tools Documents Help
Open Save Print... Undo Redo Cut Copy Paste Find Replace
nsaved Document 1
#67: "REAL PROGRAMMERS DON'T WRITE DOCUMENTATION" - BY SALVATORE IOVENE, NOV. 12TH, 2008
HTTP://WWW.GEEKHEROCOMIC.COM/

```
int
this_function_connects_the_handler_to_the_interrupt_signal_and_retur
ns_TRUE_in_case_of_success(
    int the_signal_we_want_to_connect,
    int the_detail_of_the_signal_just_mentioned,
    HandlerFunc the_function_we_want_to_connect,
    void * the_data_we_want_to_pass_to_the_function)
{
    int
iterator_of_the_loop_of_possible_signal_details_that_we_use_because_
we_need_to_make_sure_we_inspect_each_and_every_case;
    switch(the_signal_we_want_to_connect)
    {
        case UNINITIALIZED_SIGNAL: return FALSE;
        case HARDWARE_INTERRUPT:
        now_really_connect_the_hardware_interrupt_signal
(the_function_we_want_to_connect,
the_data_we_want_to_pass_to_the_function);
        break;
        case SYSREQ_INTERRUPT:
        for
(iterator_of_the_loop_of_possible_signal_details_that_we_use_because
_we_need_to_make_sure_we_inspect_each_and_every_case = 0;
iterator_of_the_loop_of_possible_signal_details_that_we_use_because_
we_need_to_make_sure_we_inspect_each_and_every_case <
MAXIMUM_NUMBER_OF_SIGNAL_DETAILS;
iterator_of_the_loop_of_possible_signal_details_that_we_use_because_
we_need_to_make_sure_we_inspect_each_and_every_case++)
        {
interpret_the_sysreq_interrupt_because_we_dont_wanna_do_it_here_and_
mess_up_the_code(the_detail_of_the_signal_just_mentioned,
the_function_we_want_to_connect,
```

Real programmers don't need to write Documentation.


I DON'T GET YOUR CODE. WHAT ARE THESE LINES FOR?
geek & poke
I HAVE NO IDEA. BUT IT DOES NOT WORK WITHOUT THEM
THE ART OF PROGRAMMING - PART 2: KISS

Do you understand your codes?

Comment your Codes!

What do the codes do?

```
main() {
int n, c, first = 0, second = 1, next;
cout << "How many numbers would you like to see";
cin >> n;

cout << "First " << n << " terms of the series are :- " << endl;
for ( c = 0 ; c < n ; c++ )  {
if ( c <= 1 ) next = c;
else { next = first + second; first = second; second = next; }
cout << next << endl; }
return 0;
}
```

How about this?

```
// This function display the terms in a Fibonacci sequence up to the chosen number        //
// The next number is found by adding up the two numbers (first and second) before it. //
main() {
        int numTerm;                    //number of terms
        int first = 0, second = 1;      // first 2 number in sequence start with 0 and 1
        int next;                       // next number is sum of 2 previous numbers

        cout << "How many numbers would you like to see";             // Do we need?
        cin >> numTerm;

        cout << "First " << numTerm << " terms of the series are :- " << endl;        // How about here?

        for (int i = 0 ; i < numTerm ; i++ )  {
            if ( i <= 1 ) next = i;             // for first 2 terms
            else {
                next = first + second;
                first = second;                 // move second to first
                second = next;                  // move next to second. Optional?
            }
            cout << next << endl;           // output next
        }
        return 0;
}
```

Uncommented Code

- You don't know what it's for.
- You don't know what was the logic.
- It can *take you hours to read it*.
  - After taking over the development…..

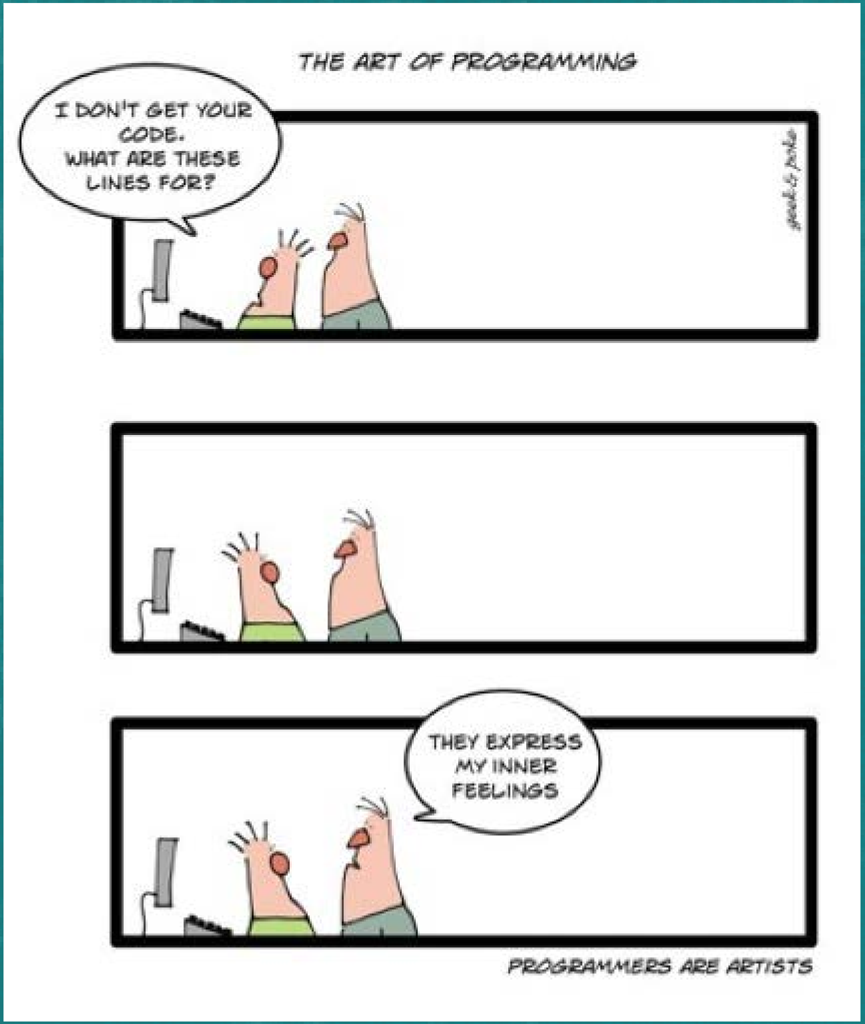
- Complications:
  - Meaningless functions/variable names
  - Poor readability (spacing, line returns, indentation)

Programmer can Read Code

- Any programmer can read code (or a block of codes)
  - have a reasonable idea of what it will do.
- But, for large amounts of code,
  - the designs and intentions implicit in how code is written and by who
  - What codes is missing
  - how the code will evolve over time

Commenting Your Codes

- Use comments to describe the intent, algorithmic overview, and logical flow.
- Especially when multiple parties are involved in modifying and maintaining code.
- Provide comments so that others could understand the behavior and purpose of the code.


THE ART OF PROGRAMMING
I DON'T GET YOUR CODE. WHAT ARE THESE LINES FOR?
THEY EXPRESS MY INNER FEELINGS
PROGRAMMERS ARE ARTISTS
geek & poke

If comments is good, then having zillions of comments in your code must be great,

right?

The Fact about Commenting Codes

- Developers are an expensive resource and documenting the code manually requires too much precious developer time
- Commenting can be a daunting task and very often postponed or even avoided.
- However, NOT documenting the code often ends up being even worse, we still have a problem.

Tools to the rescue...?

- Automatic Documentation Generation!!
  - Doxygen
  - Javadoc
  - Eclipse

- Problem Solved?
  - still have to do a bit of work and in particular...
  - You have to place some comments and/or some special tags at strategic places in your code...

Programmer’s good practices

- Give your variables, constants, functions and classes suitable names
  - const int MinutesPerHour = 60; //
  - int total = 0; // total for ?
- Provide good indentation and extra blank lines
  - Code Grouping
  - Formatting make your code look organised and easier to read.
- Function should be short, simple and do one thing.
  - A well-named function is easier to understand.
- Use comments to help other understand why you're doing
  - If you have difficulty commenting your code, chances are you that you codes are badly written.

Obfuscated Code

- …is source or machine code that has been made difficult to understand for humans.
- You don’t know what it’s for.
- You don’t know what was the logic.

```
#include <stdio.h>main(t,_,a)char *a;{return!0<t?t<3?main(-79,-13,a+main(-87,1-_, main(-
86,0,a+1)+a)):1,t<_?main(t+1,_,a):3,main(-94,-27+t,a)&&t==2?_<13? main(2,_+1,"%s %d %d\n"):9:16:t<0?t<-72?main(_,t,
"@n'+,#'/*{}w+/w#cdnr/+,{}r/*de}+,/*{*+,/w{%+,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#\;#q#n+,/+k#;*+,/'r :'d*'3,}{w+K
w'K:'+}e#';dq#'l \q#'+d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n'){)#}w'){){nl]'/+#n';d}rw' i;# \){nl]!/n{n#'; r{#w'r nc{nl]'/#{l,+'K
{rw' iK{;[{nl]'/w#q#n'wk nw' \iwk{KK{nl]!/w{%'l##w#' i; :{nl]'/*{q#'ld;r'}{nlwb!/*de}'c \;;{nl'-
{}rw]'/+,}##'*}#nc,',#nw]'/+kd'+e}+;#'rdq#w! nr'/ ') }+}{rl#'{n' ')# \}'+}##(!!/") :t<-50?_==*a?putchar(31[a]):main(-
65,_,a+1):main((*a=='/')+t,_,a+1) :0<t?main(2,2,"%s"):*a=='/'||main(0,main(-61,*a, "!ek;dc i@bK'(q)-
[w]*%n+r3#l,{}:\nuwloca-O;m .vpbks,fxntdCeghiry"),a+1);}
```

Must Read (will be quizzed)

- /* You Are Expected to Understand This */
  - http://freecode.com/articles/you-are-expected-to-understand-this
- 13 Tips to Comment Your Code
  - http://www.devtopics.com/13-tips-to-comment-your-code/
- Top 15+ Best Practices for Writing Super Readable Code
  - http://net.tutsplus.com/tutorials/html-css-techniques/top-15-best-practices-for-writing-super-readable-code/

Collaborative Development

Collaborative Tools

Collaborative Tools

- Useful when many people work on a project.
  - Across geographical boundaries
- A standard is used to create the documentation.
- A repository for the team to share documentations.
- Most importantly, it allows several persons to update it at the same time.

Common Tools

- File hosting service (e.g. Dropbox, Google Drive)
- Wikis (e.g. Wikipedia)
  - A website that allows the easy creation and editing of any number of interlinked web pages via a web browser using a simplified markup language or a WYSIWYG text editor.
- Collaborative Document Editors (e.g. GoogleDocs)
  - Documents, presentation, worksheet etc

Benefits on Online Collaboration

- Anytime, anywhere
  - Since members may be located across geographical borders, meeting may cause loss of time and money
- Documents are all stored in a single place, members can review, comment and make changes to the document easily
  - Track resources, progress online
  - Share ideas, mind-mapping
- Keep track of revision history
  - Who and when the revision made

Barriers to Overcome

- Communication
- Security
  - Vulnerable to spies, hackers
  - Higher security cost
  - Some companies still preferred to work on site.
- Group dynamics
  - Leadership harder to establish
  - Trust among members of the team
  - messages can be easy misinterpret because lack of visual and para verbal cues.