

Comments on Comments

Admin

- a quiz coming soon

–

- Reminder

- Speakers coming. Please attend if you can.
- Mon and Thursday so far.

An old debate

- Can code be truly self documenting?
 - Do we really need comments?
 -

An old debate

- Can code be truly self documenting?
 - Do we really need comments?
- My phd advisor's take.
-

An old debate

- Can code be truly self documenting?
 - Do we really need comments?
- My phd advisor's take.
- Your book's take
 - Make code as self documenting as possible – any comments needed is a failure of clear code.
- My take:

An old debate

- Can code be truly self documenting?
 - Do we really need comments?
- My phd advisor's take.
- Your book's take
 - Make code as self documenting as possible – any comments needed is a failure of clear code.
- My take:
 - As per book, except that dynamically typed languages need a little more comment TLC.

The problem with comments in real life

- Orphaned comments (see page 54)
- Comment rot
 - Comment written – then code is changed – but not comment
 - To misquote Mark Twain:

The problem with comments in real life

- Orphaned comments (see page 54)
- Comment rot
 - Comment written – then code is changed – but not comment
 - To misquote Mark Twain: there are three kinds of lies
 - Lies! Damn Lies and old Comments!!

So what comments do you still need?

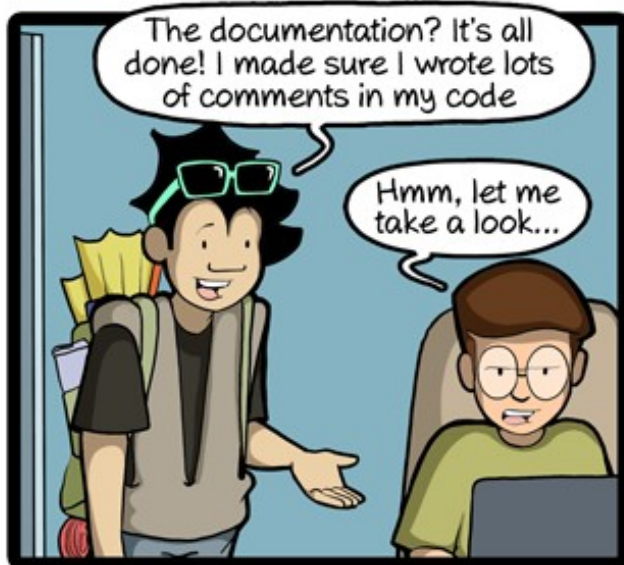
- Legal comments
 - These are barely for the code anyway
- Explaining why you chose to do it some way
- ToDo
 - These are comments meant to go away
 - explain why something doesn't make sense

So what comments do you still need?

- Warnings:
 - Warn of the consequences of code
 - Don't do this unless ...
- Informative for bizarre needs
 - Fixing a bug in a library that you don't own
example
- Documentation comments
 - Javadoc comments
 - Python API documentation strings/comments

Bad comments

- Some comments are bad
 - If someone else reading the comment has to read code elsewhere to know what the comment means
 - Book: mumbling
- Redundant comments
 - `j=j+1; //increment j`
 - Uh-huh because I failed comp151
 - Note book has larger definition
 - But awesome example on page 64 (and next slide)



Bad Comments

- Cute comments:

- Eg from stack overflow:

- `/**`

- `* For the brave souls who get this far: You are the chosen ones,`

- `* the valiant knights of programming who toil away, without rest,`

- `* fixing our most awful code. To you, true saviors, kings of men,`

- `* I say this: never gonna give you up, never gonna let you down,`

- `* never gonna run around and desert you. Never gonna make you cry,`

- `* never gonna say goodbye. Never gonna tell a lie and hurt you.`

- `*/`

Bad-ish comments

- Mandated comments
 - Some employers (DoD was famous for this) require certain structures and strictures to be followed
 - Maybe useless, but must have them
- Journal comments
 - Make a comment at top of file everytime a change is made to the code
 - This is what commit comments in the version control system are for

More bad comments

- The pasto
 - Copy – pasted code with comments
 - Change code
 - Hello comment rot.
- Commented out code
 - Now we do this in academia
 - But for code that will go to production:
 - Commented out code stays forever
 - Too important to delete?
 - Worse – multiline commented code

Bad-ish comments again

- Closing brace comments
 - Often mention matching opening brace
 - Problems?

Bad-ish comments again

- Closing brace comments
 - Often mention matching opening brace
 - Problems?
 - What if we add something new and the braces don't match up any more (comp151)
 - If we only have two levels of indent and lots of small methods, then we don't need this

Bad for many reasons

- And bad for several reasons:
 - //
 - // Dear maintainer:
 - //
 - // Once you are done trying to 'optimize' this routine,
 - // and have realized what a terrible mistake that was,
 - // please increment the following counter as a warning
 - // to the next guy:
 - //
 - // total_hours_wasted_here = 42
 - //

Don't be cute!

```
for (int i=0; i<3 /*aww*/; i++)  
{  
  █  
}
```

Same as names

- Make comments relevant
 - stop(); // Hammertime!
 - Probably was very cute 15-20 years ago
 - “You see kids...” (one day this meme will be as obsolete as your parents music)

Bee Movie? Why? (2007-2018)

```
27 #include <bits/stdc++.h> // According to all known laws of aviation, there is no
28 using namespace std; // way a bee should be able to fly. Its wings are too small
29 typedef uint16_t ul; // to get its fat little body off the ground. The bee, of
30 typedef int16_t l; // course, flies anyway because bees don't care what humans
31 // think is impossible. Yellow, black. Yellow, black. Yellow, black. Yellow,
32 #define SIZE 6000 // black. Ooh, black and yellow! Let's shake it up a little.
33 typedef bitset<SIZE> bf; // Barry! Breakfast is ready! Ooming! Hang on a second.
34 // Hello? - Barry? - Adam? - Oan you believe this is happening? - I can't. I'll
35 bf generate(ul size){ // pick you up. Looking sharp. Use the stairs. Your father
36 // paid good money for those. Sorry. I'm excited. Here's the graduate.
37 res.flip(); // We're very proud of you, son. A perfect report card, all B's.
38 res[0] = false; // Very proud. Ma! I got a thing going here. - You got lint on
39 ul increment = 3; // your fuzz. - Ow! That's me! - Wave to us! We'll be in row
40 // 118,000. - Bye! Barry, I told you, stop flying in the house! - Hey, Adam. -
41 ul limit = (ul)sqrt((double)size); // Hey, Barry. - Is that fuzz gel? - A
42 while(increment < limit) { // little. Special day, graduation. Never thought
43 for(ul i = increment*increment; i < size; i += increment*2) { // I'd make
44 res[i/2] = false; // it. Three days grade school, three days high school.
45 // Those were awkward. Three days college. I'm glad I took a day and
46 // hitchhiked around the hive. You did come back different. - Hi, Barry. -
47 do { // Artie, growing a mustache? Looks good. - Hear about Frankie? - Yeah.
48 increment += 2; // - You going to the funeral? - No, I'm not going.
49 } while(increment < limit && !res[increment/2]); // Everybody knows, sting
50 // someone, you die. Don't waste it on a squirrel. Such a hothead. I guess
51 return res; // he could have just gotten out of the way. I love this
52 // incorporating an amusement park into our day. That's why we don't need
53 // vacations. Boy, quite a bit of pomp... under the circumstances. - Well,
54 bf r; // Adam, today we are men. - We are! - Bee-men. - Amen! Hallelujah!
55 // Students, faculty, distinguished bees, please welcome Dean Buzzwell.
56 inline bool is_prime(const ul q) { // Welcome, New Hive Oity graduating class
57 return (q==2 || ((0x1 & q) && (r[q/2] ))); // of... ..9:15. That concludes
58 // our ceremonies. And begins your career at Honex Industries! Will we pick
59 // ourjob today? I heard it's just orientation. Heads up! Here we go. Keep your
60 bool is_happy(ul q) { // hands and antennas inside the tram at all times. -
61 // Happy number :D // Wonder what it'll be like? - A little scary. Welcome to
62 if (q == 1) { // Honex, a division of Honesco and a part of the Hexagon Group.
63 return true; // This is it! Wow. Wow. We know that you, as a bee, have
64 // worked your whole life to get to the point where you can work for your
65 // whole life. Honey begins when our valiant Pollen Jocks bring the nectar to
66 // Very unhappy numbers :( // the hive. Our top-secret formula is
67 if (q == 4 || q == 16 || q == 37 || q == 58 || // automatically
68 q == 89 || q == 145 || q == 42 || q == 20) { // color-corrected,
69 return false; // scent-adjusted and bubble-contoured into this soothing
70 // sweet syrup with its distinctive golden glow you know as... Honey! - That
71 // girl was hot. - She's my cousin! - She is? - Yes, we're all cousins. -
72 ul sum {0}; // Right. You're right. - At Honex, we constantly strive to
73 ul tmp; // improve every aspect of bee existence. These bees are
74 // stress-testing a new helmet technology. - What do you think he makes? - Not
75 tmp = q%10; // enough. Here we have our latest advancement, the Krelman. -
76 sum += tmp*tmp; // What does that do? - Oatches that little strand of honey
77 q /= 10; // that hangs after you pour it. Saves us millions. Oan anyone work
78 // on the Krelman? Of course. Most bee jobs are small ones. But bees know that
79 --- dank.cpp<happyprime> L28 <N> (C++/l company WK ivy FlyC:1/0 Pro
```

```
23 q /= 10; // that hangs after you pour it. Saves us millions. Oan any
24 // on the Krelman? Of course. Most bee jobs are small ones. But bees k
25 tmp = q%10; // every small job, if it's done well, means a lot. Bu
26 sum += tmp*tmp; // carefully because you'll stay in the job you pick
27 q /= 10; // rest of your life. The same job the rest of your life?
28 // know that. What's the difference? You'll be happy to know that be
29 tmp = q%10; // species, haven't had one day off in 27 million years. S
30 sum += tmp*tmp; // just work us to death? We'll sure try. Wow! That
31 q /= 10; // mind! "What's the difference?" How can you say that?
32 // forever? That's an insane choice to have to make. I'm relieved. Now
33 tmp = q%10; // have to make one decision in life. But, Adam, how co
34 sum += tmp*tmp; // never have told us that? Why would you question a
35 q /= 10; // We're bees. We're the most perfectly functioning society o
36 // You ever think maybe things work a little too well here? Like what?
37 tmp = q%10; // one example. I don't know. But you know what I'm talkin
38 sum += tmp*tmp; // Please clear the gate. Royal Nectar Force on approa
39 q /= 10; // a second. Oheck it out. - Hey, those are Pollen Jocks! - W
40 // never seen them this close. They know what it's like outside the hiv
41 return is_happy(sum); // but some don't come back. - Hey, Jocks! - Hi
42 // You guys did great! You're monsters! You're sky freaks! I love it
43 // it! - I wonder where they were. - I don't know. Their day's not
44 // Outside the hive, flying who knows where, doing who knows w
45 int main(int argc, char *argv[]) // can'tjust decide to be a Pollen J
46 { // have to be bred for that. Right. Look. That's more pollen than y
47 ios_base::sync_with_stdio(false); // will see in a lifetime. It'
48 cin.tie(NULL); // status symbol. Bees make too much of it. Perhaps
49 // you're wearing it and the ladies see you wearing it. Those ladies
50 r = generate(10000); // they our cousins too? Distant. Distant. Look
51 ul count; // two. - Oouple of Hive Harrys. - Let's have fun with them.
52 cin >> count; // be dangerous being a Pollen Jock. Yeah. Once a bear p
53 for (ul i = 0; i < count; ++i) { // against a mushroom! He had a p
54 ul k, candidate; // throat, and with the other, he was slapping m
55 cin >> k >> candidate; // my! - I never thought I'd knock him out. W
56 cout << k << ' ' // you doing during this? Trying to alert the auth
57 << candidate // I can autograph that. A little gusty out ther
58 << (is_prime(candidate) // wasn't it, comrades? Yeah. Gust
59 && is_happy(candidate) ? " YES\n" : " NO\n"); // h
60 } // sunflower patch six miles from here tomorrow. - Six miles, huh?
61 // A puddle jump for us, but maybe you're not up for it. - Maybe I a
62 return 0; // are not! We're going 0900 at J-Gate. What do yo
63 // buzzy-boy? Are you bee enough? I might be. It all depends on v
```

And the “grand prize”

- Be careful, someday your irritation might get you on the front page too:
- RichardIsAFuckingIdiotControl
 - // The main problem is the BindCompany() method,
 - // which he hoped would be able to do everything. I hope he dies.
- <http://mcfunley.com/from-the-annals-of-dubious-achievement>

Assignment

- Read chapter 4 in clean code.