James Moorhead
CS 160

Dreaming In Code: Answers

Chapters 0-3

CH. 0

1. Donald Knuth, author of programming textbooks
2. Zero
3. Dedicated/obsessed computer programmer
4. 2/3rds of projects
5. Frederick P. Brooks, Jr

CH. 1

1. Michael Toy-Project Manager
   John Anderson-Lead Coder
   Ted Burgess-Programmer
   Mitchell Kapor-Founder Funder
   Lou Montulli-Programmer
2. List that programmers keep their known list of bugs
3. Open Source Application Foundation
4. Chandler Project
5. Personal Information Manager-Handle things like calendars, to-do lists, e-mails, etc
6. Scary
7. Not knowing how long could take to fix
8. Software time
9. Term for slight lateness on a project
10. IBM system/360
11. Adding manpower to a late software project makes it later
12. $1/6^{th}$
13. 1/2
14. Only when a task can be partitioned among many workers with no communication between them
15. Source code is the editable code used to make changes or view how a program is assembled.
    A program on your computer has been ran through a compiler as to be able to be read by your computer aka executable or .exe

16. Making software anyone can tinker with or apply changes or fixes if wanted ie “open source” code
17. Good Programmers may try to do everything themselves in a new or different way Great Programmers know when to reuse or rewrite and build off that
18. Bazar being quick releases often and mass contribution with different methods Cathedral being slow small group whole release
19. Yes
20. Work on stuff you enjoy in the project and let the rest follow

CH. 2

1. That if all software is a closely guarded secret it will die out
2. The original term being “bootstrapping” Meaning: an improving of the improvement process
3. Xerox Palo Altos Research Center
4. Virtual Case File Project
   Vaporware
   Everest

CH. 3

1. “Because you almost certainly won’t get it right the first time.”
2. A core dump is when the computer stops what it’s doing and dumps all its active memory at the time of the crash into a file as to be looked over later to discover what the problem was. Cores more referring to when memory banks were built using wound wired coils known as ferrite cores.
3. Assembly Language
4. Compiler

CH. 4

1. Front end: The side of the program the user sees and interacts with.
   Back end: Where user input and results from front end are stored and executed
2. Being able to buy coded software that can “snap” together no coding needed to make a functioning program from those “blocks”
3. Most programming languages don’t follow the same strict structure and programming itself doesn’t follow an exact formula and so there’s not enough uniformity to it.

CH. 5

1. Speed, Price and Quality
2. A more widely used term for obsessing over something specifically.
3. Rewriting code to make it briefer clearer and easier to read without changing it functionally.
4. “seemingly pointless activity which is actually necessary to solve a problem which solves a problem which, several levels of recursion later, solves the real problem you were working on.” Eric Raymond

CH. 6

1. Areas of software that may contain bugs that passed through testing and into the end product not found until many people have bought and used the product resulting in all possible scenarios to play out and missing bugs to be found.
2. Don’t start with a large project start with a small project and never expect it to get big otherwise you’ll overdesign it.

CH. 7

1. At its basics it’s a method/ scheme of naming variables and objects.
2. “Communicating abstractions unambiguously-from programmer to machine, from programmer to programmer and from programmer to user.”

CH. 8

1. Use your own products as to help find bugs, help maybe potentially spark further ideas and give the end user perspective as the programmer.
2. This would be around the time of chandler 0.4. And it was completed “the traditional thing that engineers do is make them up.”