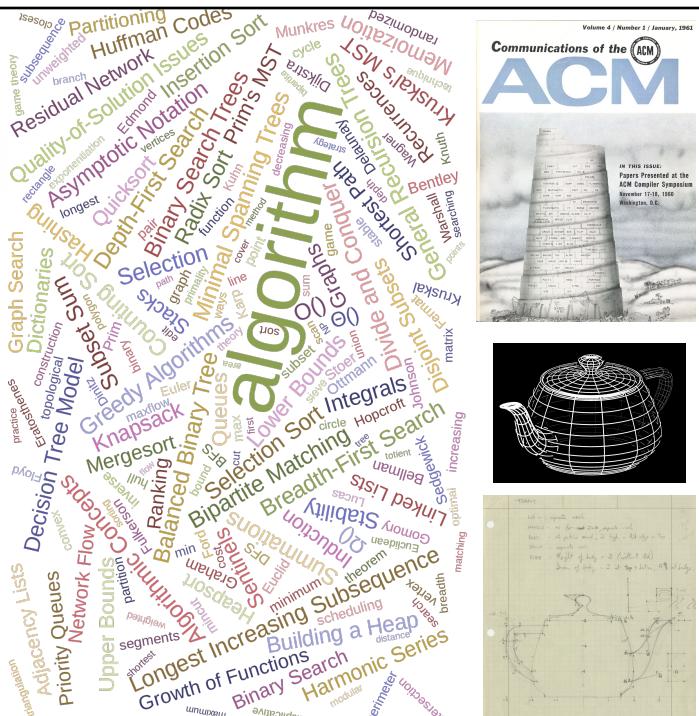
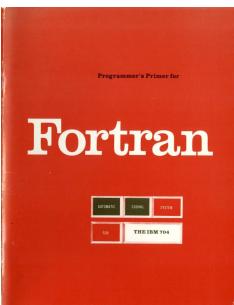
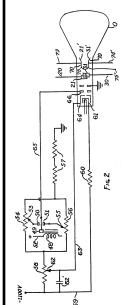


```

(reproduce animal)) *animals*) (add-plants))(defun draw-world () (loop for y below *height* do
animal-y animal) y)) *animals*) #'M) ((gethash (cons x y) *plants*) #'*) (t #\space))) ((pr
str :junk-allowed t))) (if x (loop for i below x do (update-world)) if (zerop (mod i 1000) d
*player-strength* nil)(defparameter *monsters* nil)(defparameter *monster-builders* nil)(defi
monsters-dead) (princ nil))(defun game-loop () (unless (or (player-dead) (monsters-dead)) (
fresh-line) (map 'list (lambda(m) (or (monster-dead m) (monster-attack m))) *monsters*)) (*gam
*player-health* 0))(defun show-player () (fresh-line) (princ nil) (princ *player-health*) (p
(monster-hit (pick-monster) (+ 2 (randval (ash *player-strength* -1)))) (let ((i (randval
monster-hit (pick-monster) x)))) (otherwise (dotimes (x (1+ (randval (truncate (/ *player-st
mons* -1) (random 1 (* (max (min monster-level (1+ (random 10 100)) (random 10 100)) (if
(fresh-line) (princ (list (cadr (acons (acons (acons (acons (acons (acons (acons (acons (acons
(princ *monsters*) (defparameter monster-builders (randval (random 10 100)) (defparameter mons
:include monster) (push #'make-hydra *monster-builders*)(defmethod monster-show ((m hydra)
(princ nil)) (princ (princ nil) (princ (type-of m))) de
monster-show ((m orc)) (princ nil) (princ (orc-club-level m)) (princ nil)) (defmethod monster
:include monster) (push #'make-hydra *monster-builders*)) (defmethod monster-show ((m hydra)
(princ nil)) (princ (princ nil) (princ nil))) (defmethod monster-attack ((m hydra)
defstr
slime-
#*make
*playe
63) (x
monste
new-mdps
collect (cond ((member p monsters) (cond ((= p pos) (return-from main (layer-lose))
(cons (code-char (parse-integer (coerce (list (cadr lst) (caddr lst)) "string")) r
(jun
list) "string)) (defun parse-params (s) (let* ((l (position #\= s)) (i (position #\=
i2))) ((equal s nil) nil) (t s))))(defun parse-url (s) (let* ((url (subseq s (+ 2 (pos
'(cons url '))))(defun get-header (stream) (let* ((s (read-line stream)) (n (let ((i (1

```



Facts of Life

How to Succeed

Every Day I Wake Up Thinking ...

1. Oh, boy, I get to do even *more* computer stuff today!
2. Computer stuff is interesting, but I'm not *that* excited.
3. Whatever. Doing computer stuff will be an OK job.
4. At least computer stuff is better than a poke in the eye with a pointed stick.
5. Just go ahead and poke me with that stick right now.

Every Day I Wake Up Thinking ...

- If you can't answer **1** or **2**, maybe you ought to rethink your major.
 - Seriously!
- High tech is demanding and those who aren't enthusiastic about their work are not going to succeed.
 - They end up getting disconnected, feel oppressed, and ultimately burn out.
- Success in your career demands that you stay current and if you're not enthusiastic you're not going to want to keep learning.
 - It's not a 9-to-5 kind of career.

How to Succeed ...

- Come to class. Don't be late.
- Pay attention in class.
- Take your own notes. (Don't depend on just the notes that get posted to Canvas.)
- Ask questions if anything is unclear.
- Pay attention to the questions others ask and the answers they get.
- Try to answer any side questions posed in class or in the materials.
- Download and review any exercise as soon as it is posted. Do not procrastinate.
- Make a plan for doing each exercise. Do not procrastinate.
- Review materials as soon as they are posted. Do not procrastinate.
- Start preparing for the midterm on Day 1. Do not procrastinate.
- Start preparing for the final on Day 1. Do not procrastinate.
- And, especially, ***Read The Syllabus.***

Notice any patterns in all that advice?

Yeah, "Do not procrastinate."

How to Succeed ...

- And, especially, ***Read The Syllabus.***

**You never
told us
any of that!**

**It's in the
Syllabus**



Sending Me Email ...

- Use MavMail, **not** Canvas.
- Address all email to **brian.dalio@uta.edu**
- Include your Full Name, NetID, StudentID, Course number, Section number.
- Put all of this info in the BODY of your email, not the just the SUBJECT line. (Don't depend on my seeing the SUBJECT line of your message.)
- The response might be: (1) an individual email, (2) an email to the entire class, (3) a discussion in the next class meeting.
- Don't be insulted if (2) or (3) happens. Sometimes questions are so great I want the entire class to hear / participate in the response.
- Every question deserves an answer, but that answer might not be until the next class meeting.
- If you're not getting an answer, followup! (I might not have gotten your email.)

A NetID is three letters followed by four or five digits. It looks something like **abc1267**.

A StudentID is ten digits. It looks something like **1001123456**.

A “Bad” email ...



Fri 2019-08-23 13:45

Dalio, Brian A ✉



Hello Dr. Dalio

Unfortunately I missed the first day of class and I would like to know if i missed anything important about the class or is there any assignment that will be due any time or soon? I will be in the next class.

Thank you
Sincerely

Maddog87

Cargo Cult

"CARGO CULT"

F E Y N M A N

NO MAGIC

FORM vs.
CONTENT

So who is "Feynman"?

- Anybody know?
- Richard P. Feynman, theoretical physicist
 - Quantum mechanics, superfluidity, parton model, ...
 - Nobel prize in Physics, 1965
- Popularizer of physics
 - E.g., gave a talk in **1959** on nanotechnology, *The Feynman Lectures on Physics*, some autobiographical books, ...
 - Caltech commencement (1974) re: *Cargo Cult Science*.



Theory, Practice, Implementation

THEORY

Theory is just that -- THEORY. It's what we get from our analysis of problems and algorithms. We try to develop our theory using formal methods and as carefully as we can, but it's still just theory.

PRACTICE

Practice is when we actually develop and write an algorithm in (pseudo-) code. At this point we feel we are in control of the situation because we've not only done the analysis, we written the program.

IMPLEMENTATION

Implementation is what happens when our code makes it into a real-world environment, using real-world data, and getting real-world results. Unfortunately, what we observe at Implementation time isn't always what we expected from our Theoretical analysis.

Theory

What OUGHT to happen when we write the code and run it. We've developed an algorithm. We (think we) understand the time / space complexity.

Practice

Well, I've managed to write the (pseudo) code. (That is, I've figured out how to "instantiate" that theoretical algorithm.)

Implementation

Run the code on real-world data on a real-word machine
Hmm. Well, that didn't go as expected. Let's loop back to Practice or maybe even Theory and try again

Talk is Cheap

**TALK is CHEAP.
SHOW ME
THE CODE.**

-- Linus Torvalds

So who is “Linus Torvalds”?

- Anybody know?
- Creator and lead developer of Linux kernel.



IDEAS

Ideas are basically worth NOTHING.

(That's why "ideas" are not even considered intellectual property.)

Every week when we have this class, just listening to the questions you ask about the code I show gives me ZILLIONS of ideas of how I could do something in the future.

Those ideas are essentially WORTHLESS

UNLESS

UNLESS

UNLESS

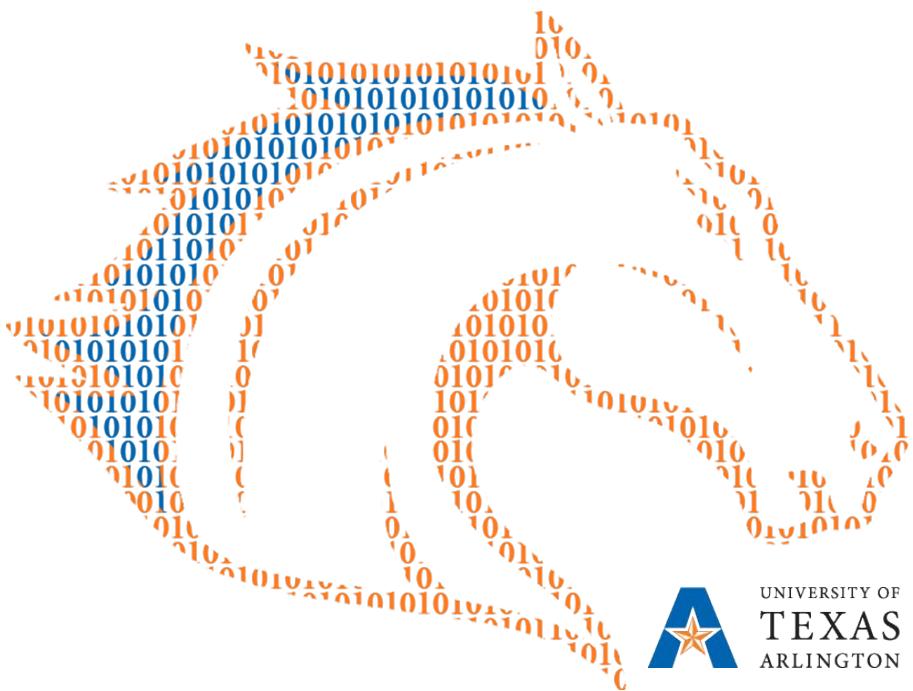
I DO SOMETHING.

TALK is CHEAP.
SHOW ME THE CODE.

-- Linus Torvalds

In the Real World, it's not the IDEA that makes the money, it's the EXECUTION.

The Real World is littered underfoot with millions of Ideas waiting to be picked up but Execution is what breathes life into them.



UNIVERSITY OF
TEXAS
ARLINGTON

DEPARTMENT OF
COMPUTER SCIENCE
AND ENGINEERING