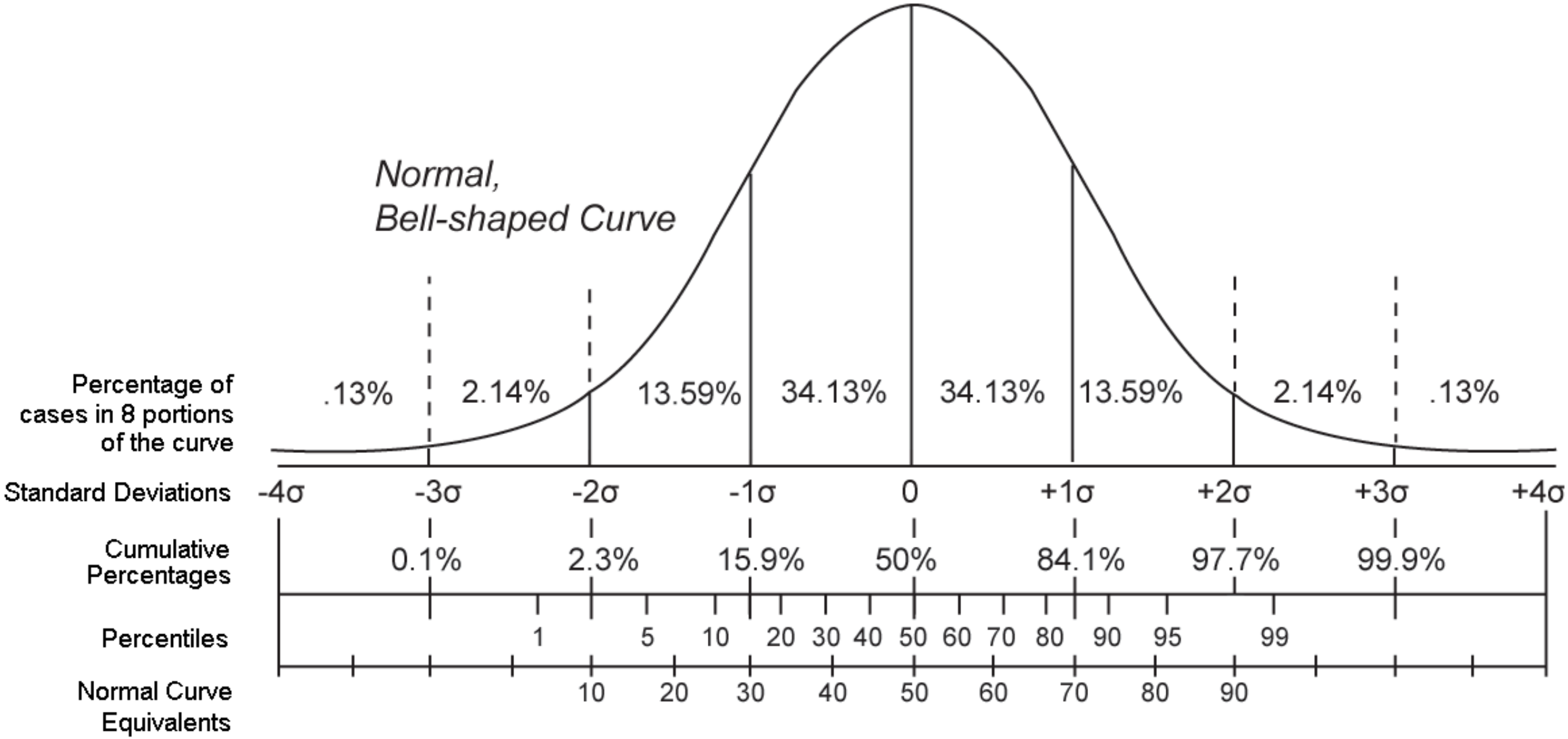
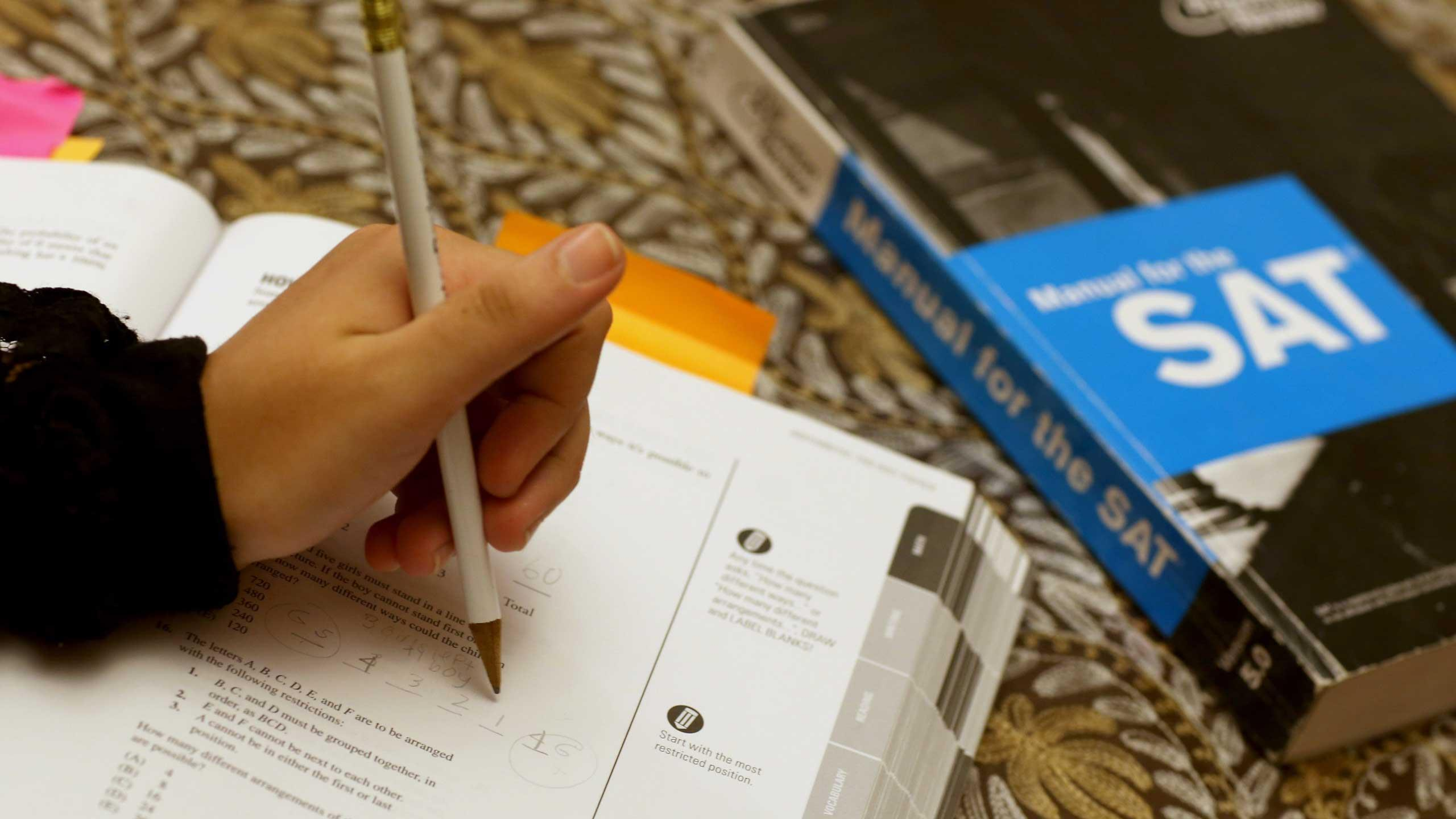
University Skills

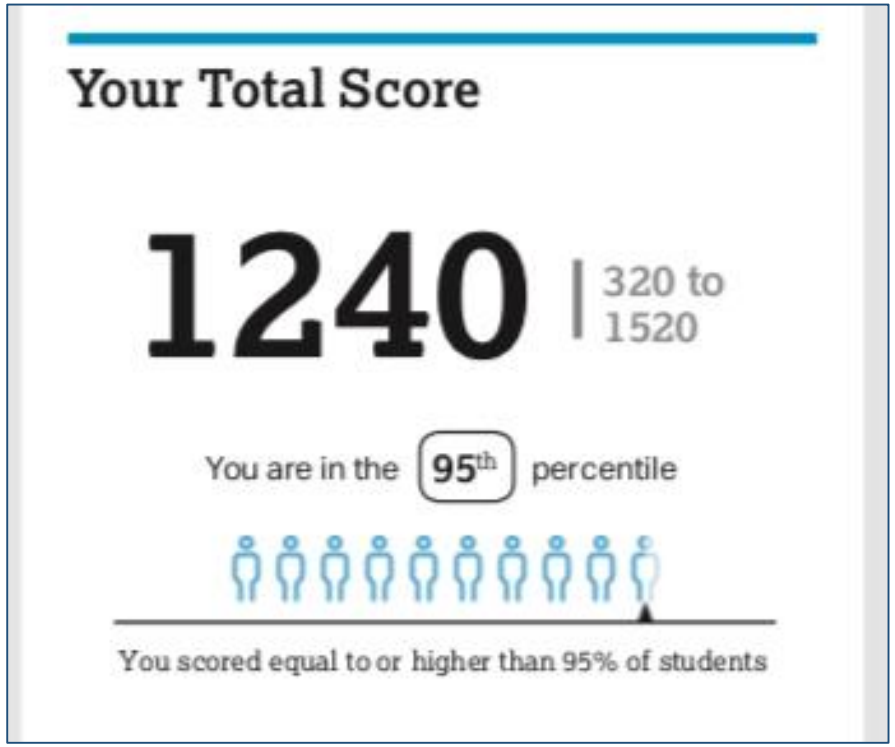
Ebbinghaus

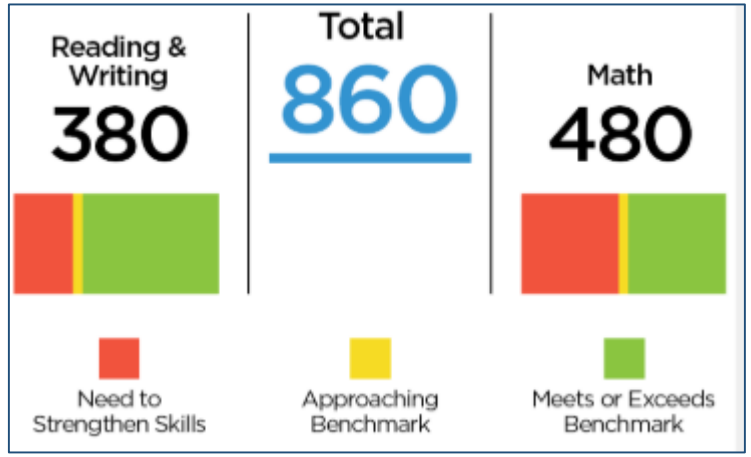

Normal,
Bell-shaped Curve
Percentage of cases in 8 portions of the curve
.13%
2.14%
13.59%
34.13%
34.13%
13.59%
2.14%
.13%
Standard Deviations
-4σ
-3σ
-2σ
-1σ
0
+1σ
+2σ
+3σ
+4σ
Cumulative Percentages
0.1%
2.3%
15.9%
50%
84.1%
97.7%
99.9%
Percentiles
1
5
10
20
30
40
50
60
70
80
90
95
99
Normal Curve Equivalents
10
20
30
40
50
60
70
80
90


Manual for the SAT
Start with the most restricted position.
VOCABULARY

SAT scores use percentiles

SAT Composite Range	Percentile Score	Equivalent Letter Grade
1530-1600	99+	A+
1450-1530	99 to 99+	A+
1420-1450	98 to 99	A
1360-1420	95 to 98	A
1310-1360	92 to 95	A-
1270-1310	88 to 92	B+/A-
1210-1270	82 to 88	B+
1160-1210	76 to 82	B-/B
1120-1160	70 to 76	C
1060-1120	60 to 70	D
1000-1060	48 to 60	D
960-1000	40 to 48	F
910-960	31 to 40	F
850-910	21 to 31	F
800-850	14 to 21	F
750-800	9 to 15	F
630-750	1 to 9	F
620-630	1- to 1	F
620 and below	1-	F





Term Dean's Honours List

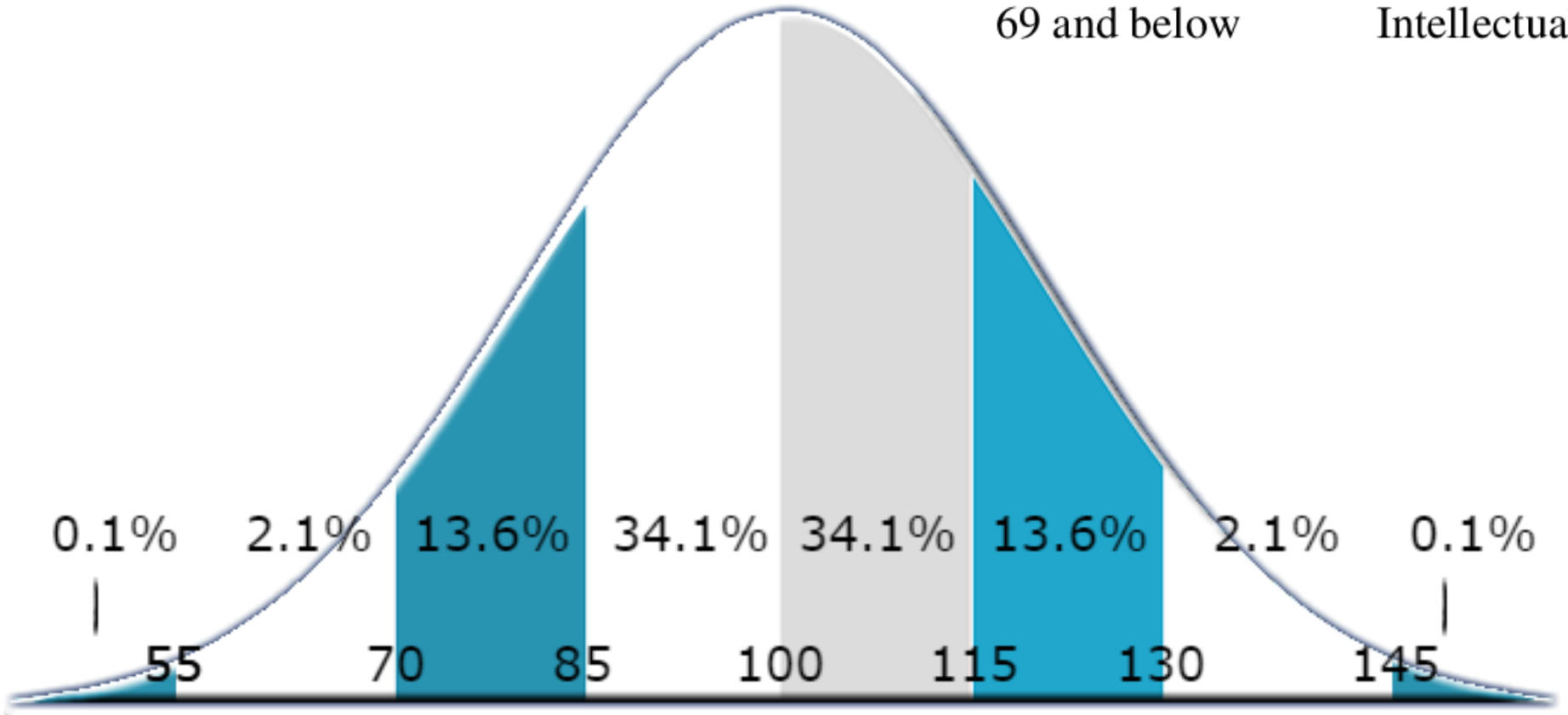
In recognition of outstanding academic achievement in an academic term, the designation "Term Dean's Honours List" is awarded to undergraduate Mathematics students who satisfy all of the following criteria for the term:

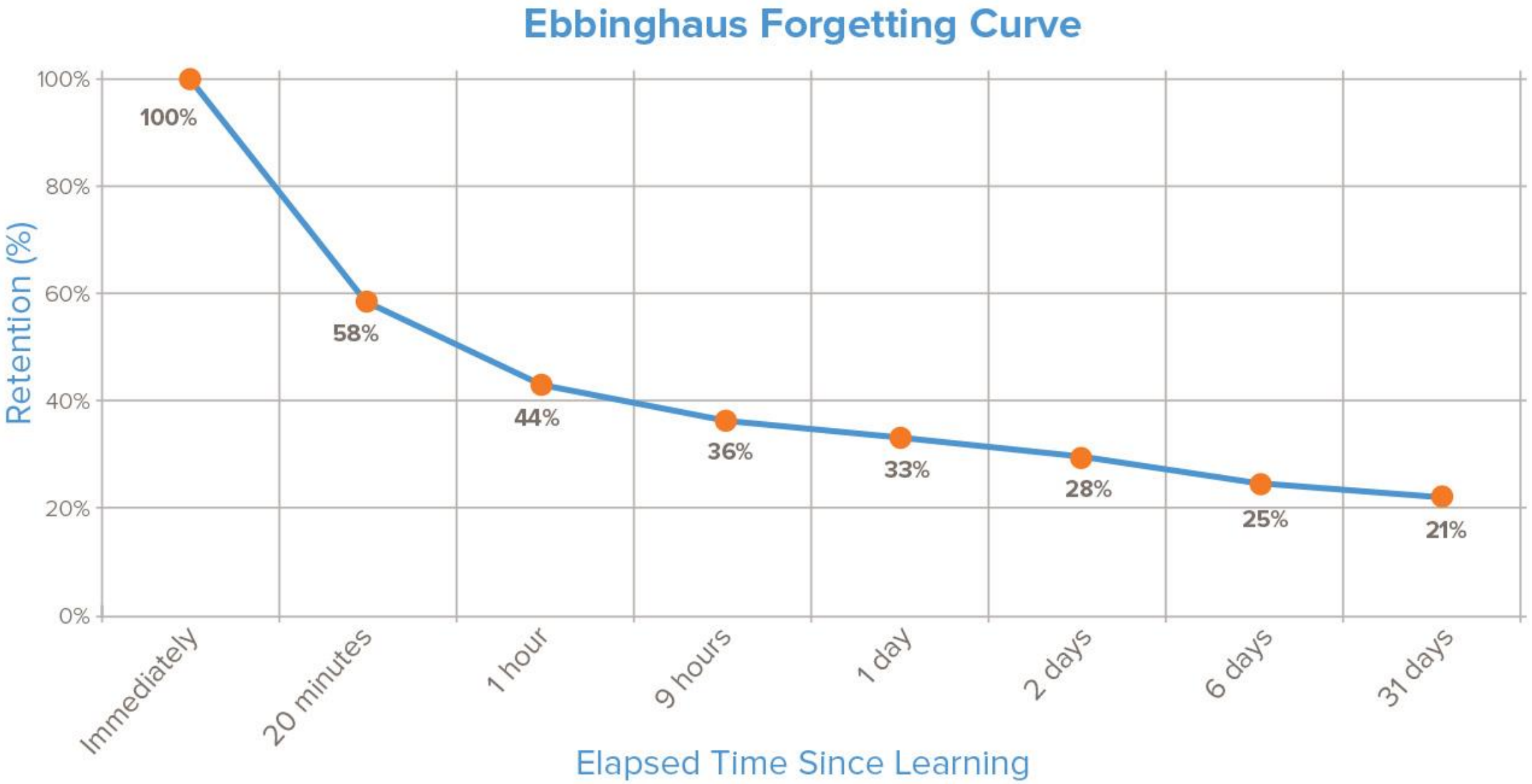
- registered in an honours plan with a Term Average (TAV) of at least 87%;
- normally enrolled in at least 2.5 units of courses with numeric or letter grades;
- no excluded courses; and
- no INC, IP, or UR grades.

But we used to say that only the top percentile (99^{th}) made the honours list.

In 2014, about 72 out of 6939 made the list which was 1%.

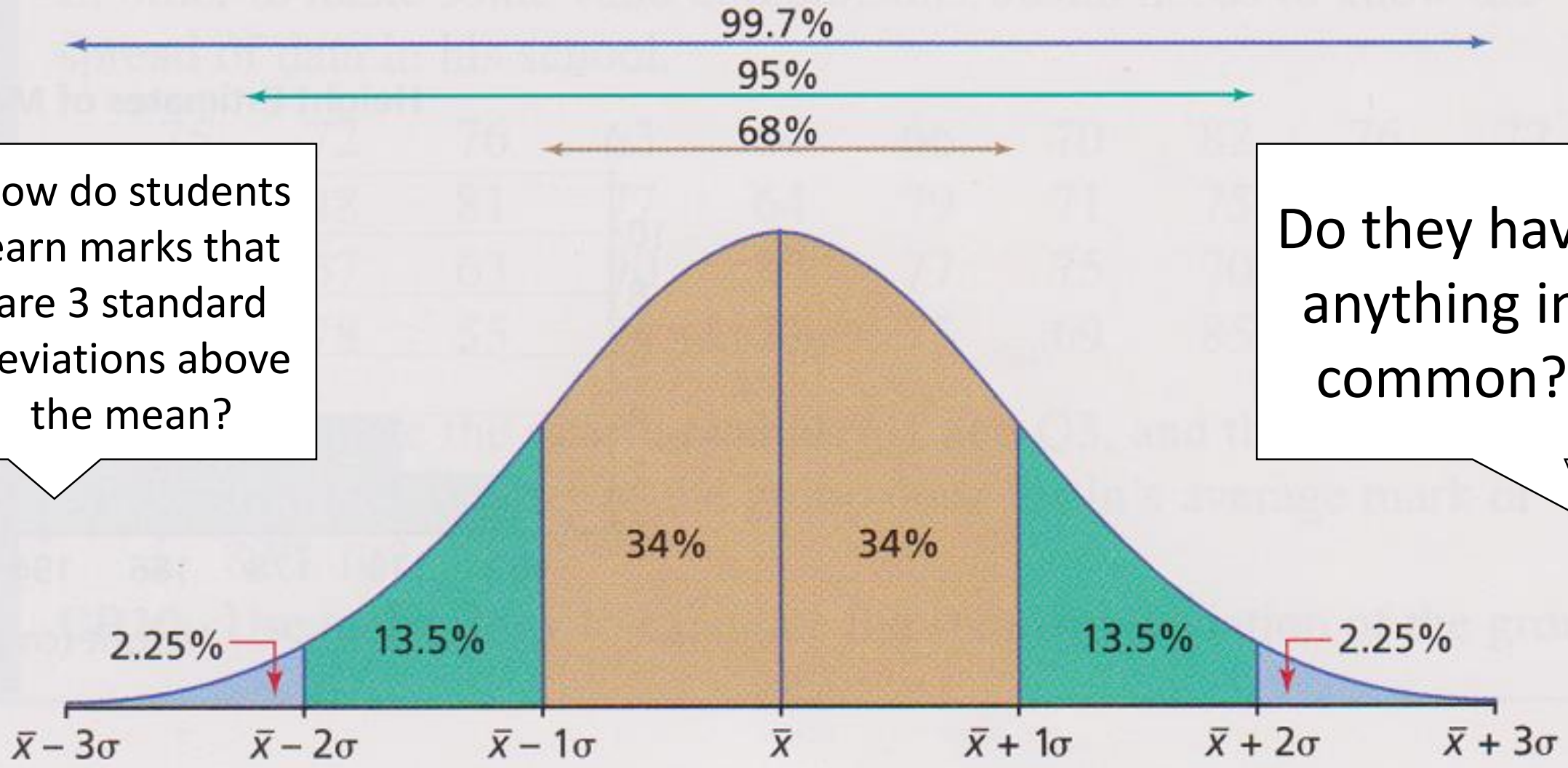
Full Scale IQ	Classification	Percentile
130 and above	Very gifted	98 – 99.5
120-129	Gifted	91 – 97
110-119	High Average	75 – 90
90-109	Average	25 – 73
80-89	Below Average	9 – 23
70-79	Border Line	2 – 8
69 and below	Intellectually Poor	0.01 – 2




Ebbinghaus Forgetting Curve
Retention (%)
100%
80%
60%
40%
20%
0%
100%
58%
44%
36%
33%
28%
25%
21%
Immediately
20 minutes
1 hour
9 hours
1 day
2 days
6 days
31 days
Elapsed Time Since Learning

How do students earn marks that are 3 standard deviations above the mean?

Do they have anything in common?



The graph of the normal distribution $X \sim N(\bar{x}, \sigma^2)$

After 20 minutes:
42% of learning
is lost

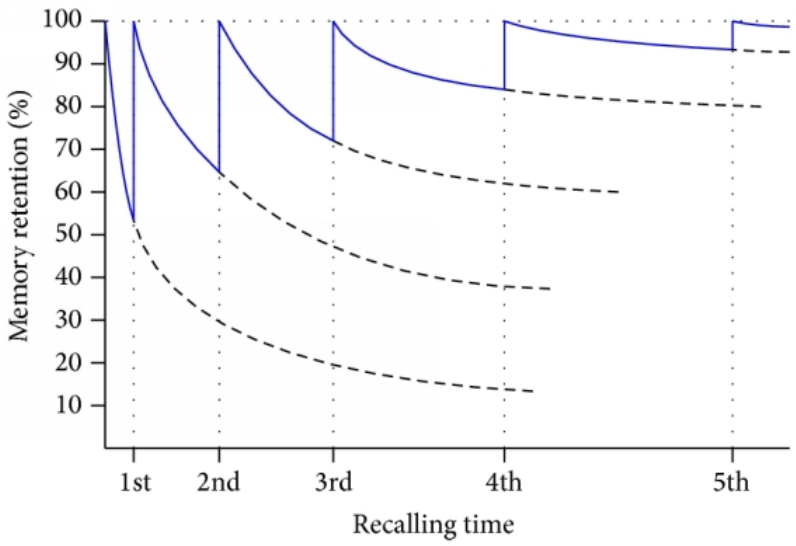
After 24 hours:
67% of learning
is lost

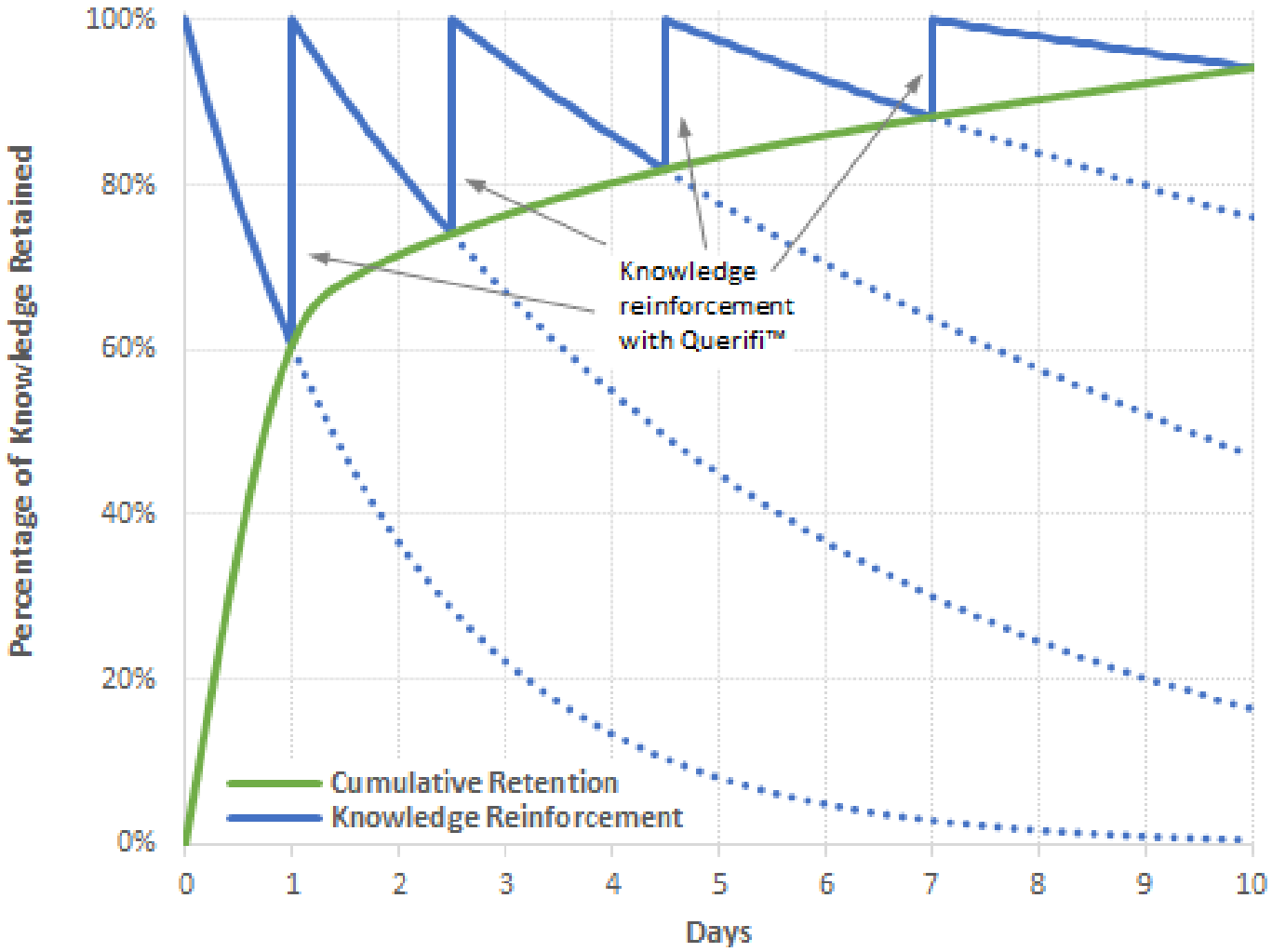
After 31 days:
79% of learning
is lost

After 60 days:
90% of learning
is lost

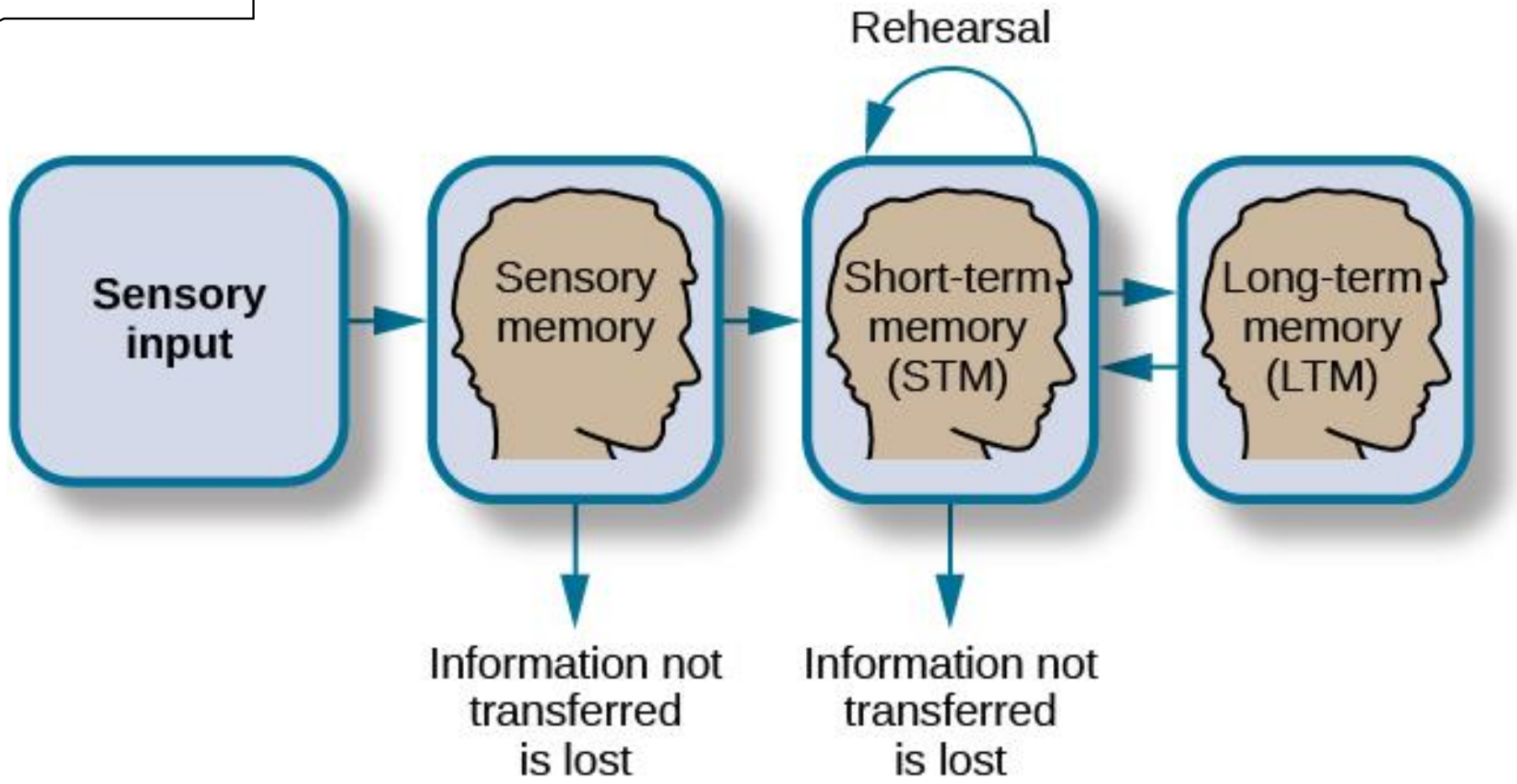
Without further revision, the average learner retains only 10% of new information after 60 days

Combatting the Forgetting Curve



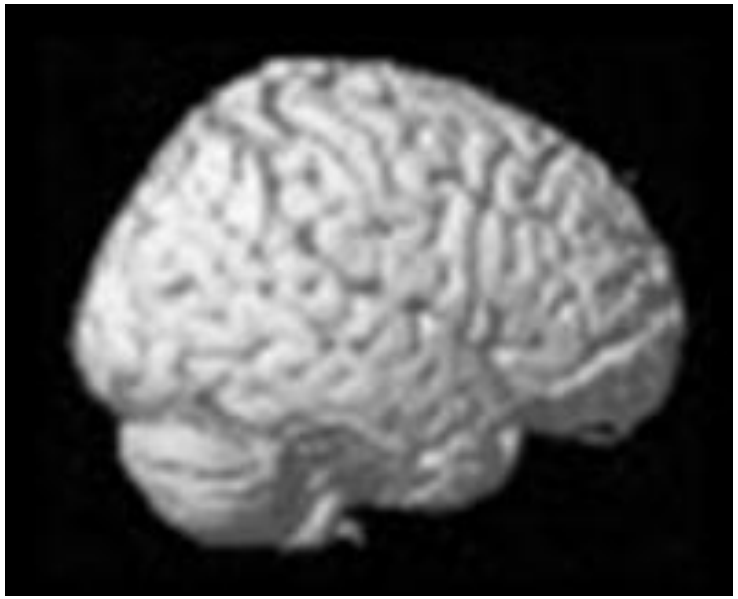


Studying

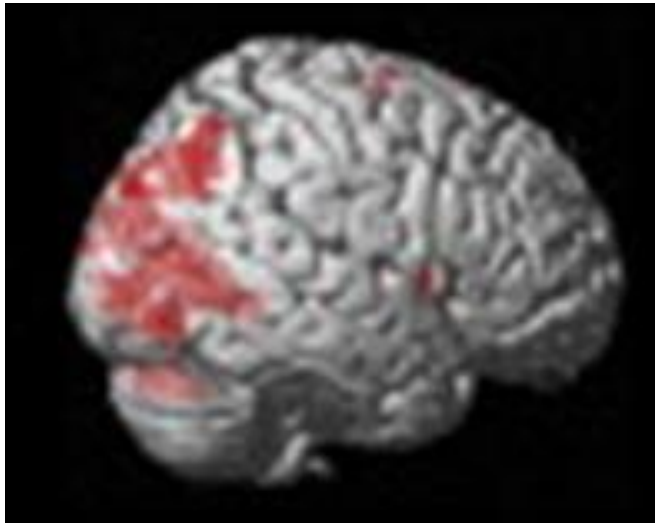

Rehearsal
Sensory input
Sensory memory
Short-term memory (STM)
Long-term memory (LTM)
Information not transferred is lost
Information not transferred is lost

Tone Discrimination Training Task

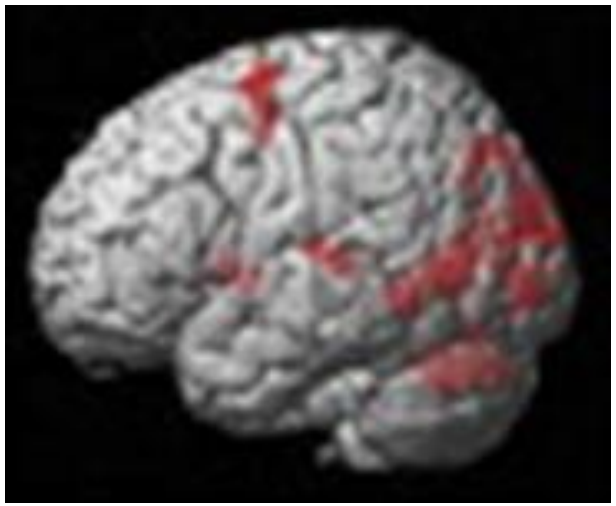
Before Training



After, Successful

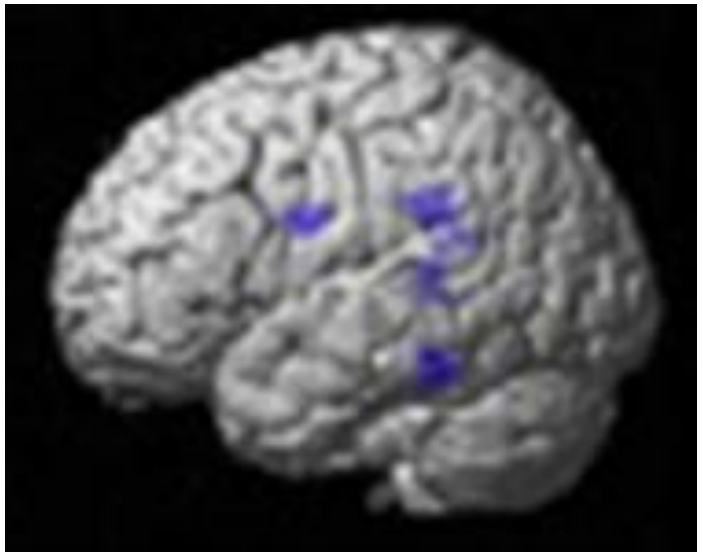


After, Not Successful

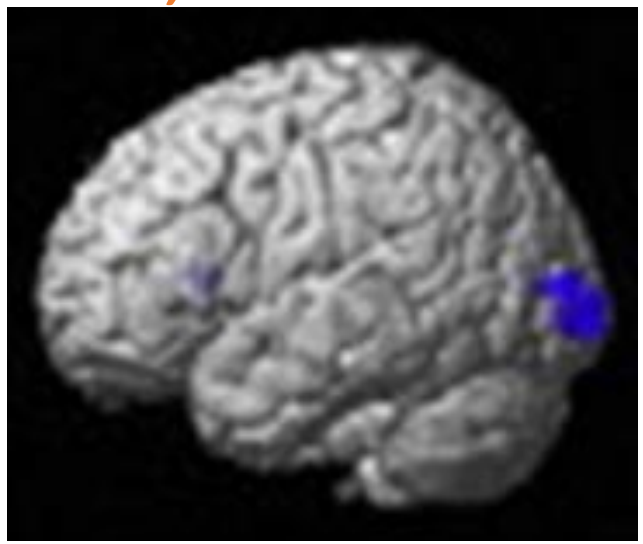


Pitch Discrimination Training Task

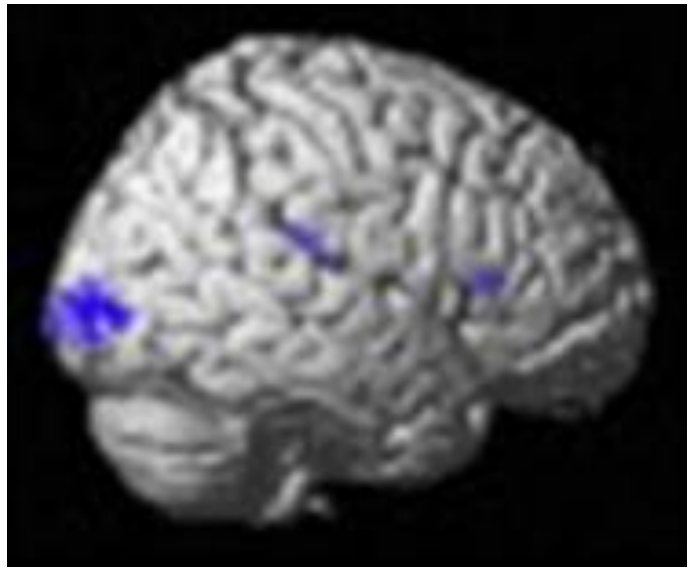
Before Training



After, Successful



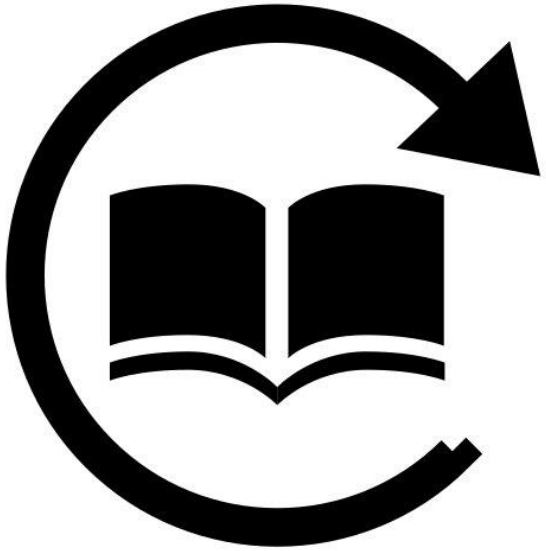
After, Not Successful



How do you study?

Highlighting, Underlining

Re-reading notes

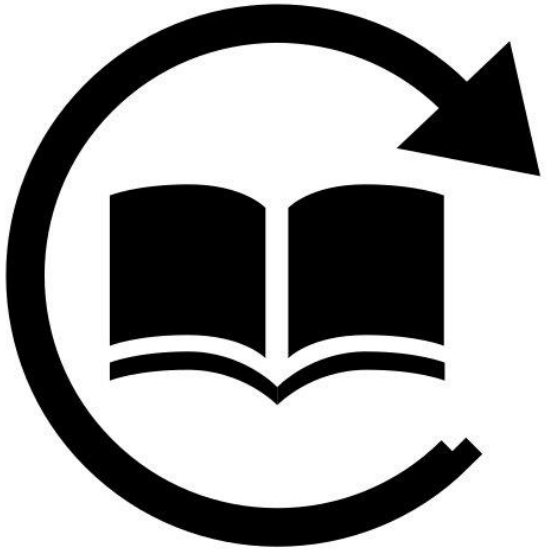


Testing yourself

How do you study?

Highlighting, Underlining

Re-reading notes

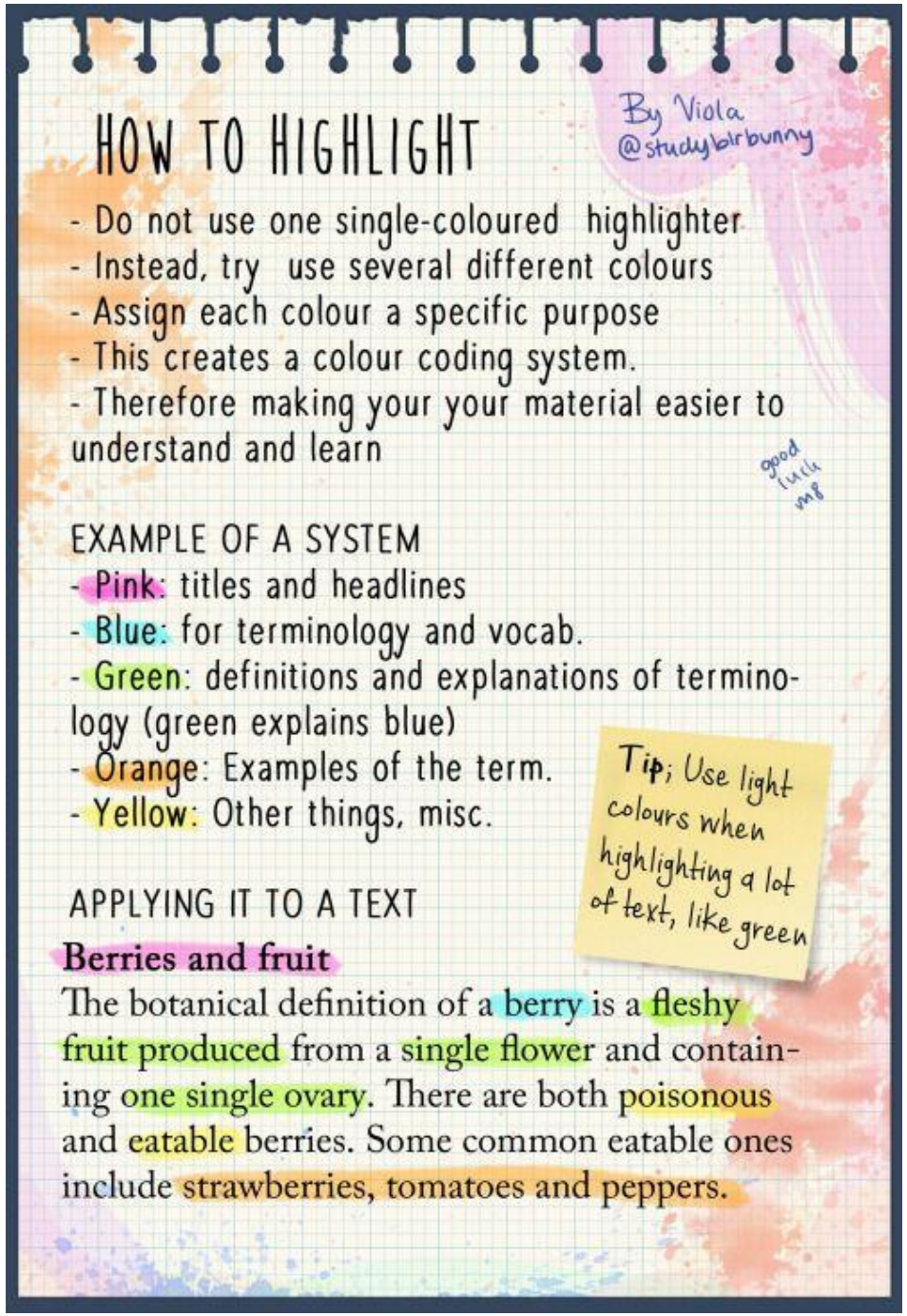


Testing yourself

84% of students do this.

The most common technique.

Highlighting



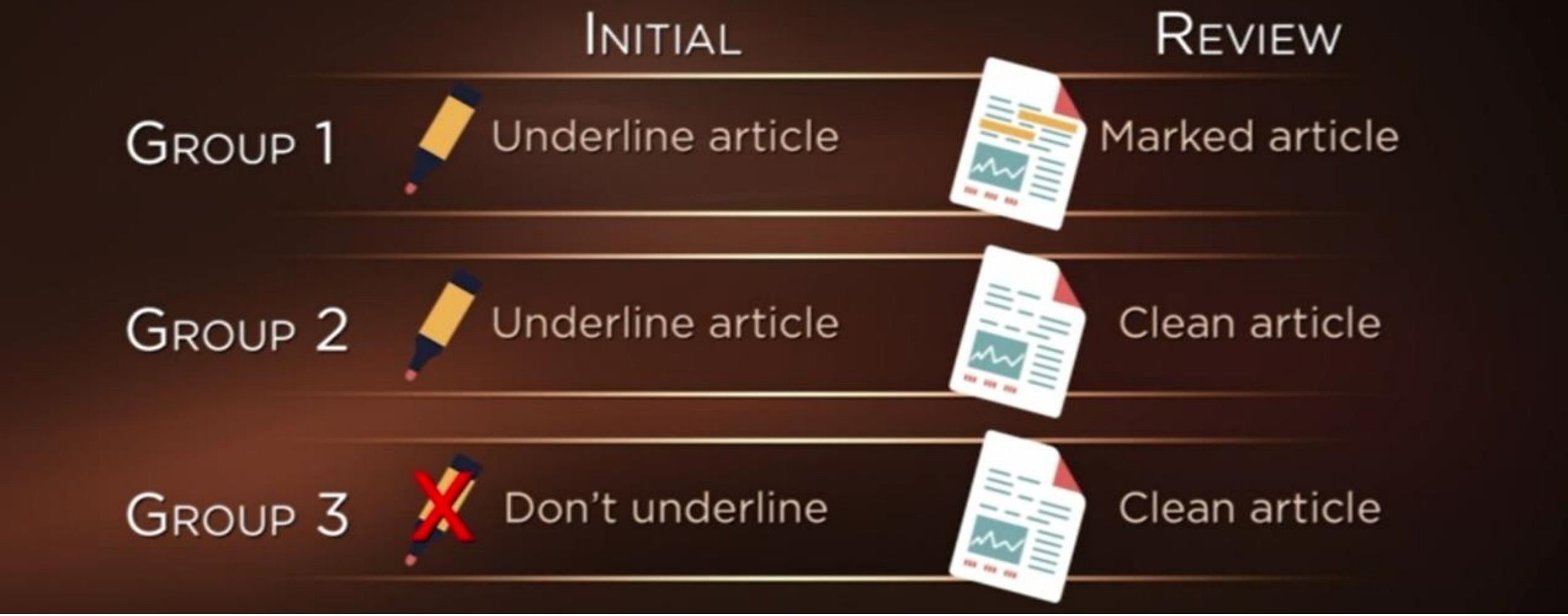
Highlighting: Peterson

	Initial	Review
Group 1	Underline article	Marked article
Group 2	Underline article	Clean article
Group 3	Don't underline	Clean article

Who did best on knowledge based questions?

On Application?

Highlighting: Peterson



Knowledge Based: Everyone did the same.

Inference Based: Group 3 did far better than Group 1 & 2

Ernest Roth-kopf. Bell Labs.

ROTHKOPF

Students were given a written passage with some of the key words missing, which they were asked to fill in.

GROUP 1

Some of the students had never seen the passage before,

ROTHKOPF

Students were given a written passage with some of the key words missing, which they were asked to fill in.

GROUP 1

GROUP 2

Some of the students had never seen the passage before, another group had read the passage once before,

ROTHKOPF

Students were given a written passage with some of the key words missing, which they were asked to fill in.

GROUP 1

GROUP 2

GROUP 3

Some of the students had never seen the passage before,
another group had read the passage once before,
a third group had read the passage twice,

ROTHKOPF

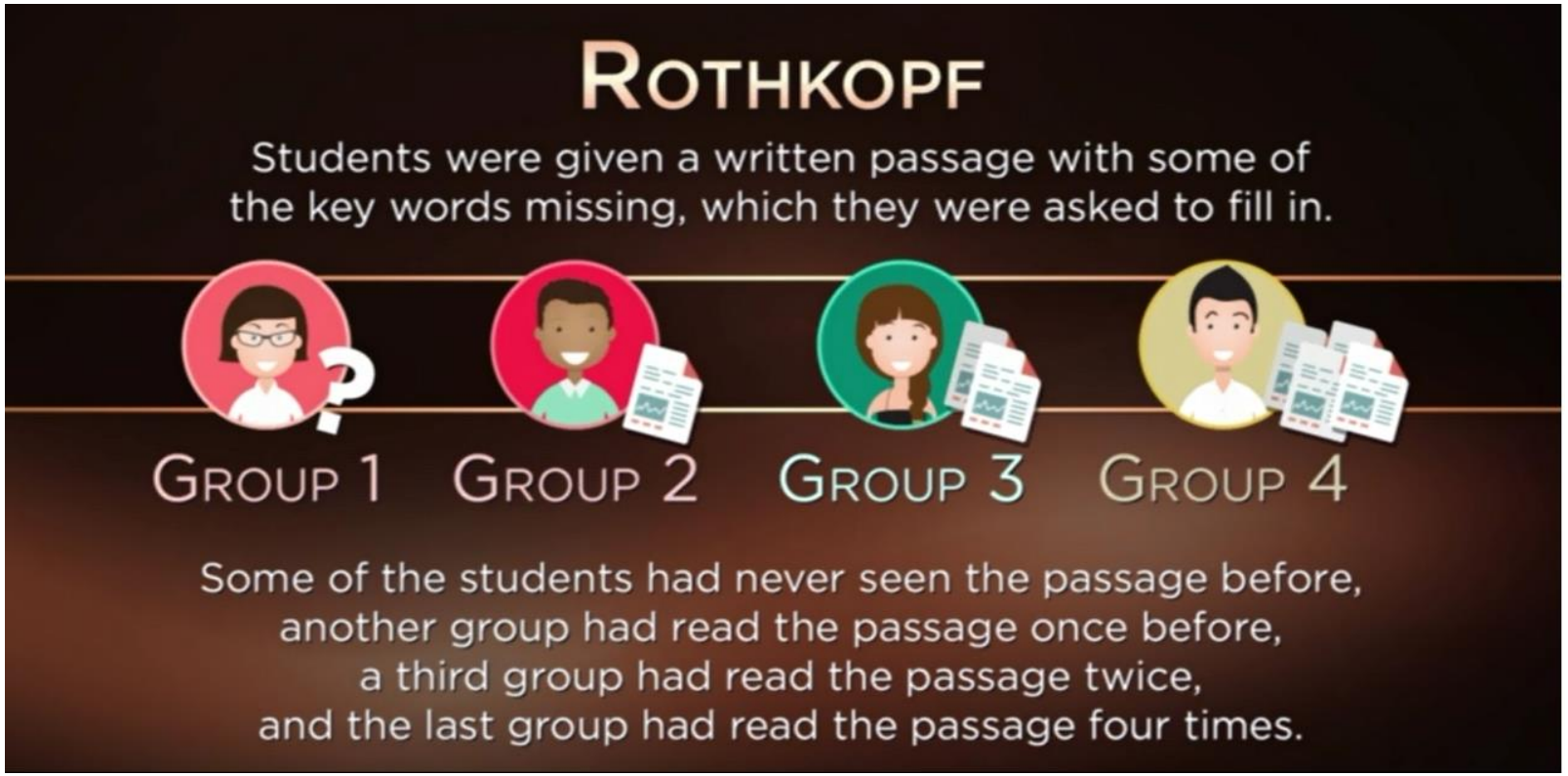
Students were given a written passage with some of the key words missing, which they were asked to fill in.

GROUP 1 GROUP 2 GROUP 3 GROUP 4

Some of the students had never seen the passage before, another group had read the passage once before, a third group had read the passage twice, and the last group had read the passage four times.

Who did best on fill-in-the blank questions?

On Knowledge?
On Application?

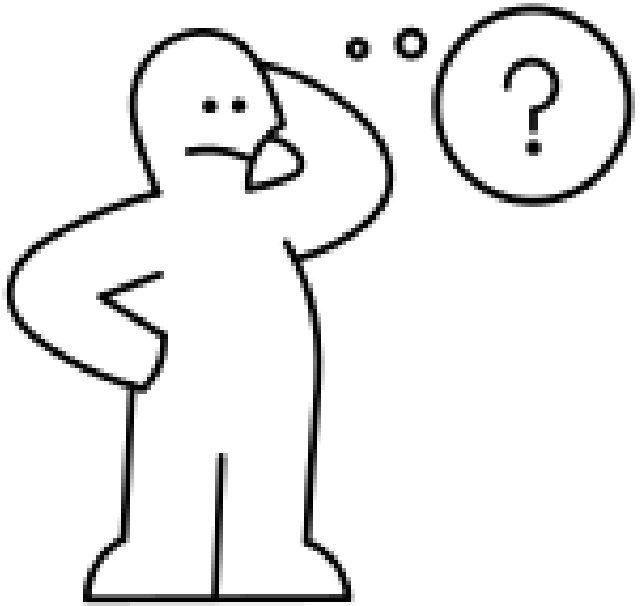


Group 3 – 10% of more on fill-in-the-blanks.
After that, no difference. Group 4, no difference.

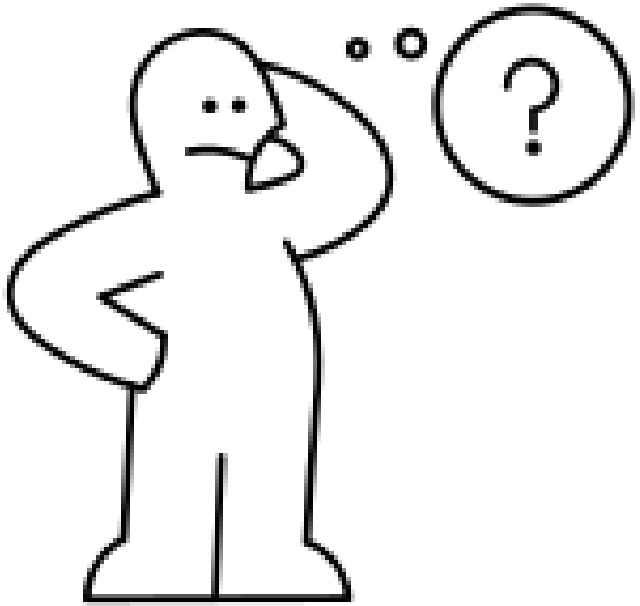
Only works for fill-in-the blanks. Not inference, knowledge, application based.

Re-reading a passage 2, 3, 4 times adds no value.
You get a sense that it is familiar. You can recognize information that you've seen before.
But recalling and recognizing is very different.

Who were the last 3 vice presidents of the United States?

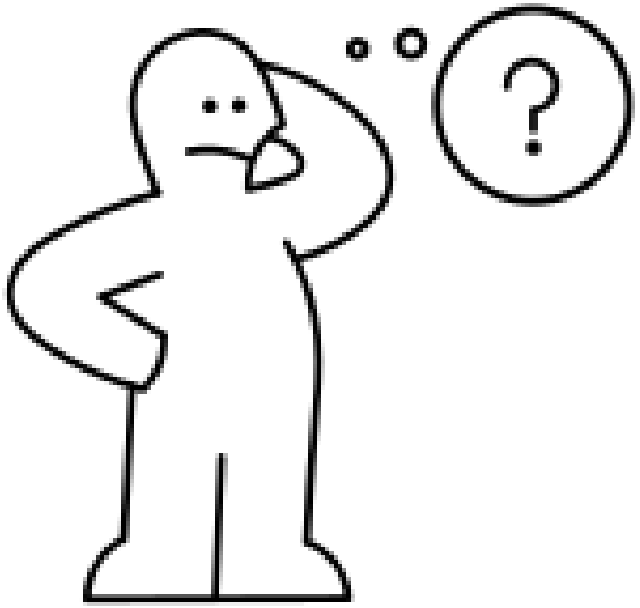


Who were the last 3 vice presidents of the United States?



Dick Cheney
John McCain
Elizabeth Warren
Mike Pence
Nancy Pelosi
Mitt Romney
Joe Biden

Who were the last 3 vice presidents of the United States?



Dick Cheney 3
John McCain
Elizabeth Warren
Mike Pence 1
Nancy Pelosi
Mitt Romney
Joe Biden 2

If you read a chapter and then re-read it, you will recognize a lot of what you read. That familiarity will fool you into thinking that you really know it.
Unfortunately, you will be in for a rude awakening when you actually write the test and need to RECALL the information. That's a more in-depth brain change.

This happens all the time – students who study by re-reading are often very surprized. They often say they aren't good at test taking. Or that tests aren't fair or useful. That's not the trouble. They just didn't change their brain.

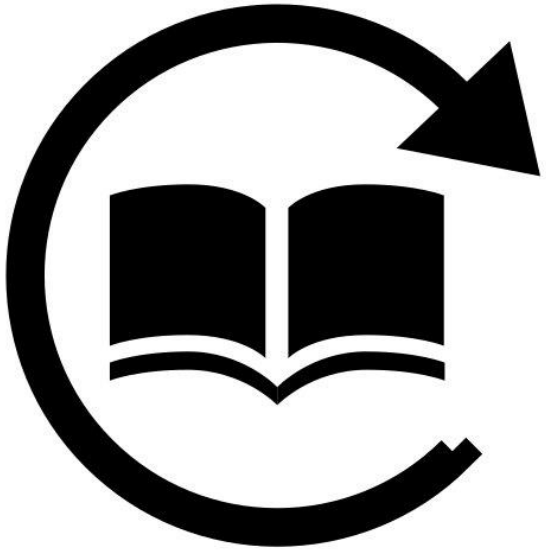
Re-reading is ok, it's not the best. Any time you spend reading isn't time you spend effectively building your brain.
I repeatedly tell my students not to bother re-reading. It's a waste of time.

The key is that memory consolidation is ACTIVE. Not passive.

How should you study?

Highlighting, Underlining

Re-reading notes



Testing yourself

Lectures

They can be boring if all that you do is listen.

There are people who advocate "understanding in lectures"

I can't process that quickly.

A form of unit testing

CS 134 Unit 1— Software construction Slide 17

Black-box testing

as a user sees code (not at all!)

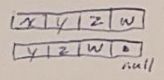
- Design test cases based on the specifications. or interfaces
- Recall: pre- and postconditions define a contract. just need method signatures & pre & post conditions, not the code.
  - test input should meet preconditions
  - test runs should compare output to expected results implied by postconditions
- "typical" input: type of data expected to be common in practice any value that is not special
- *boundary values*: data that makes the precondition(s) "barely" true also test "special" cases eg. null.

For example,

```
public void removeMe (Object[] array);
// pre:  array not null
// post: removes first occurrence of this, if any, closing
//       gap and setting the last entry to null
```

test cases involving `x.removeMe()`: remove x from the array.

[]	array.length == 0
[x]	x is the only member
[null]	null is the only member
[y,…,x,…,z]	x is in the middle
[x,…,y]	x is at the start
[y,…,x]	x is at the end
[y,…,z]	x is not in the array
[y,…,x,…,x,…,z]	x repeats



array is not null; array is size 1 and has a pointer to null - not what is in the pre condition

you don't test anything in the preconditions - that is the job of the person who calls the method AND of the Integration Tester NOT the black box tester.

testing compromise
Test boundary and special cases for one parameter with typical cases for the others

This means your test cases grow linearly not exponatially

CS 134 Unit 1— Software construction Slide 18

White-box testing

- Design test cases based on the structure of the code.
- Execute every line of code.
  - *Branch testing*: tests for each alternative
  - *Loop testing*: tests to iterate
    - 0 times
    - exactly once
    - several times
    - as often as possible
  - *Exit testing*: tests to cause each condition for loop or method exit
  - *Exception testing*: test of exception handling

coverage analysis
it is the pieces of code that are rarely used that are most likely to have a problem.

Continuing example:

```
public void removeMe (Object[] array) {
// pre:  array not null
// post: removes first occurrence of this, if any, closing
//       gap and setting the last entry to null
  int i;
  for (i = 0; i < array.length; i++) {
     if (array[i] == this) break;
  }
  if (i == array.length) return;
  while (i < array.length-1) {
     array[i] = array[i+1]
     i++;
  }
  array[i] = null;
}
```

find x

shuffle stuff in its place

white box testing shows you that you test the while loop executes exactly once [y,…, x, z] when x is the second to last occurance.

remember:
how many parameters does this method have? print()
one. the object 'this'.

I concentrated on following the examples.

I wrote down things the professor said about the slide (because I can't remember)

White-box testing

- Design test cases based on the structure of the code.
- Execute every line of code.
  - *Branch testing*: tests for each alternative
  - *Loop testing*: tests to iterate
    - 0 times
    - exactly once
    - several times
    - as often as possible
  - *Exit testing*: tests to cause each condition for loop or method exit
  - *Exception testing*: test of exception handling

coverage analysis
it is the pieces of code that are rarely used that are most likely to have a problem.

Continuing example:

```
public void removeMe (Object[] array) {
// pre:  array not null
// post: removes first occurrence of this, if any, closing
//       gap and setting the last entry to null
  int i;
  for (i = 0; i < array.length; i++) {
     if (array[i] == this) break;
  }
  if (i == array.length) return;
  while (i < array.length-1) {
     array[i] = array[i+1]
     i++;
  }
  array[i] = null;
}
```

find x

shuffle stuff in its place

white box testing shows you that you test the while loop executes exactly once

[y, ..., x, z]

when x is the second to last occurance.

These are notes from a class where I had to write down the questions too

Professor said

Professor said

Professor said

Professor said

Algorithmic Problem Solving

Horse Race

You are arranging races for 25 horses on a track that can accommodate 5 horses at a time. Each horse always runs the distance in the same time and the horses have distinct speeds. You have no stopwatch, but can make deductions from the finishing order in the races. What is the smallest number of races needed to determine the 3 fastest horses, in order?

An algorithm is a step by step sequence to solve a problem.

A solution:

Divide into groups of 5. Race and pull out first 5. .... but wait... what if the top fastest are in the first group ... we don't find the top 3.

Winners stay on (lose position 4, 5 ... bring on 2 new)

How many races?

$25 - 2 = 22$

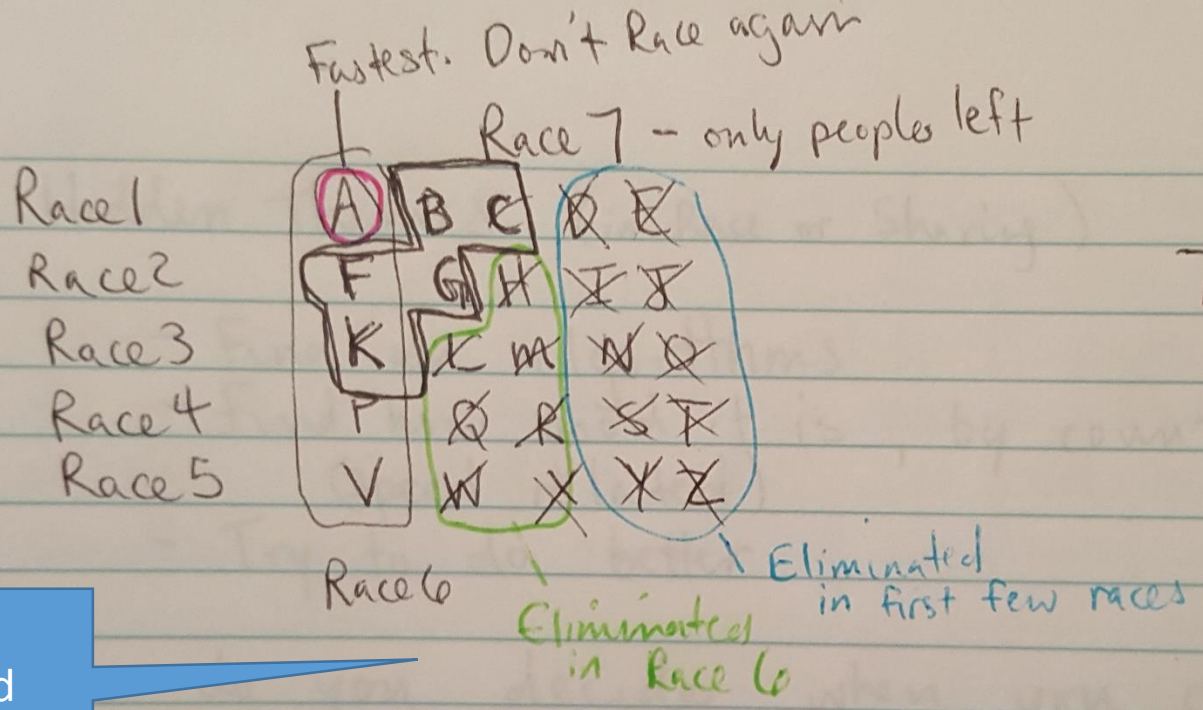
$\frac{22}{2} = 11$. 11 races are needed.

Start with groups of 5.

- Race tops together
- Race 2nds together
- Race 3rds together

Race top 3. + top 2nd + top 3.

9 races



7 Races.

End of problem, diagram on board

Sharing Information

Each of 50 people knows a different piece of information. They are allowed to exchange the information they know by phone calls between pairs of people. During any call, just one person is permitted to

At the end of the day

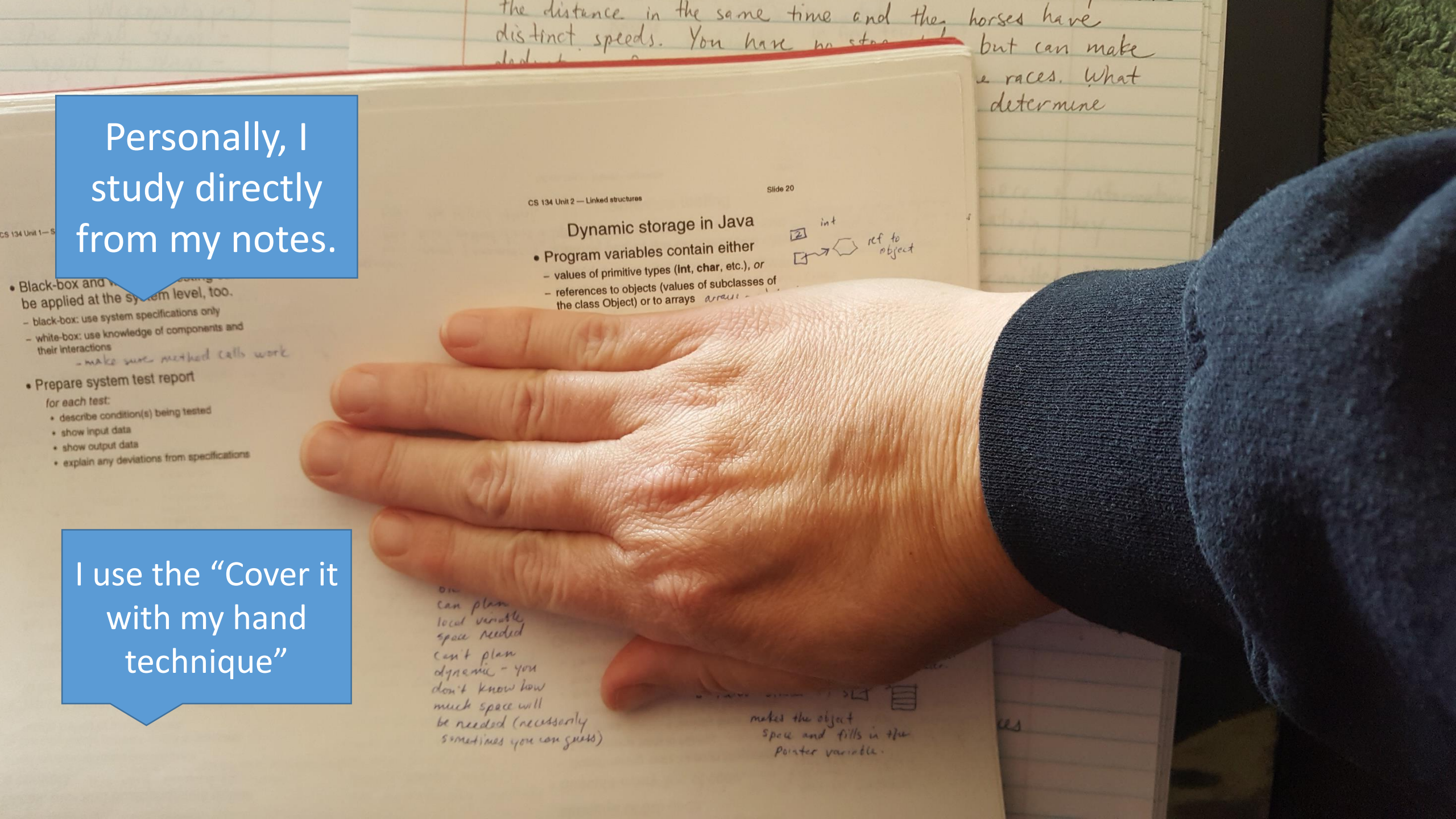
At the end of the week

At the end of the month

Rehearse the content until you can recite it back without looking

Personally, I study directly from my notes.

I use the “Cover it with my hand technique”



Slide 19

em testing

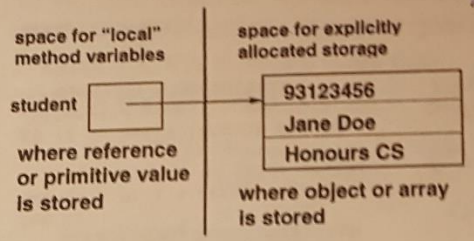
white-box testing can
he system level, too.
ystem specifications only
nowledge of components and
e sure method calls work
em test report

dition(s) being tested
lata
data
deviations from specifications

- Program variables contain either
  - values of primitive types (Int, char, etc.), or
  - references to objects (values of subclasses of the class Object) or to arrays arrays are's bit of
- Space to hold the value of a variable "local" to a method is allocated automatically. int x done when you declare it creates space for it (doesn't fill it)
- Space to hold the value of an object or array is explicitly allocated and initialized upon construction.
  - in response to use of new or array initializer done dynamically

2 ways to allocate space

Is allocated when the method ends or if declared to be in for loop; gone when for loop is done. done automatically.



a reference holds a memory address

I repeat until I don't need to peek to recall the information

local + dynamic storage is seperated b/c compiler can plan local variable space needed can't plan dynamic - you don't know how much space will be needed (necessarily sometimes you can guess)

Student s; s [?] makes space for pointer.

s = new Student() s → makes the object space and fills in the pointer variable.

the top 3.

n 2 new)

At the end of the day

At the end of the week

At the end of the month

It seems like this couldn't possibly be true, but by third & fourth year, I was studying LESS than others and getting substantially higher marks.

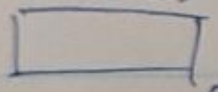
I retained everything that I had previously learned, so adding new material was quicker for me than others.

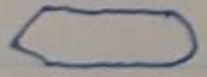
Rehearse the content until you can recite it back without looking

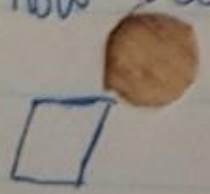
Another technique

Take notes in question-answer form.

Why are flowcharts useful?
- to plan out how your program will work
- it lays out ifs, loops, input, output before
- it's useful to visualize the order the code

How does  work in flowchart?
- it holds processes (math or variable stat
- it has one arrow coming out
- rectangle => math

How does  work in flowcharts?
- it holds output (eg. setText)
- it has one arrow coming out
- pencil shape -> writing

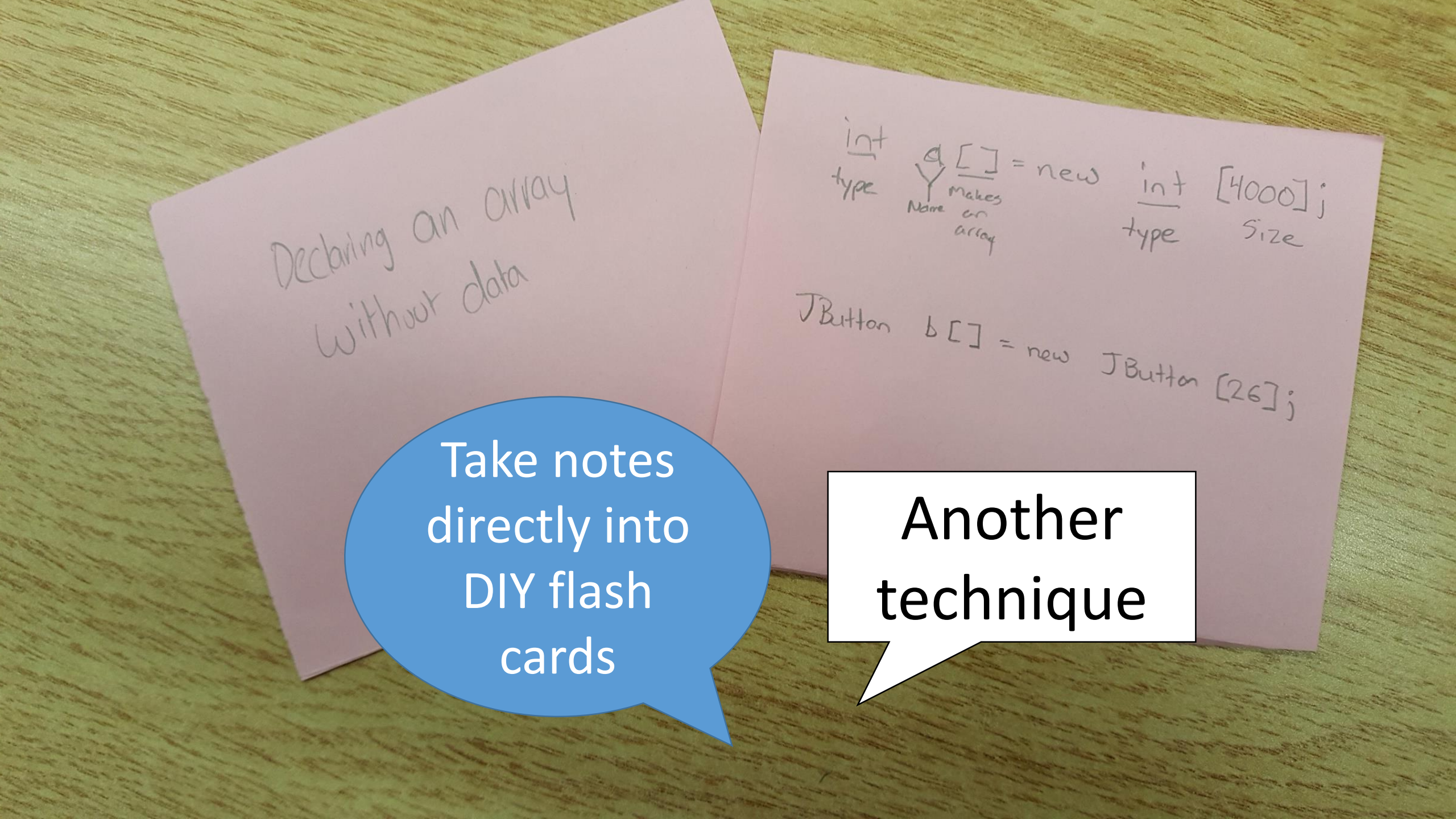
How does 
- it holds input (eg. prompt, promptNum)
- it has one arrow coming out
- it says "get variable name"

phone call

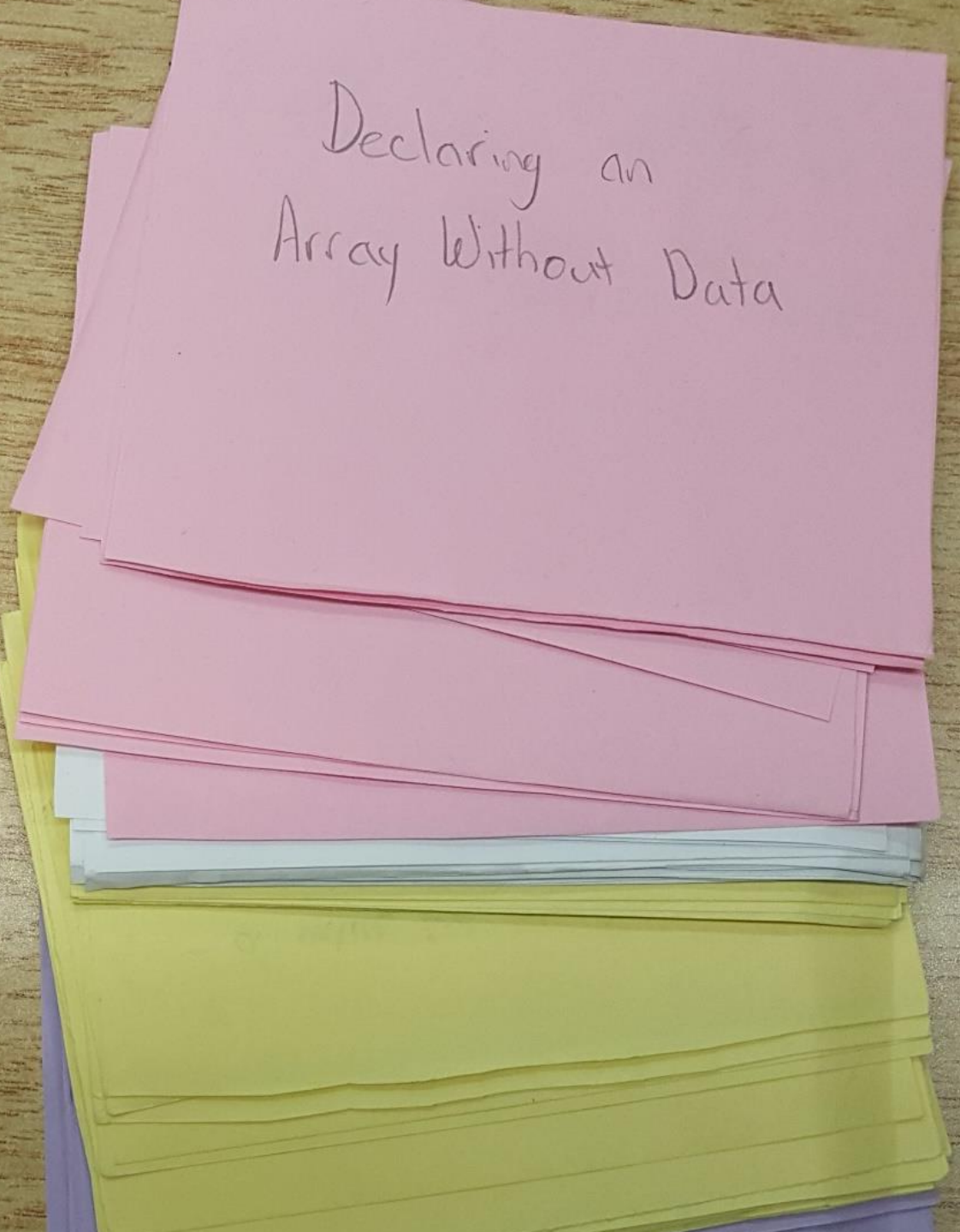
- their device requests info from servers and displays it.

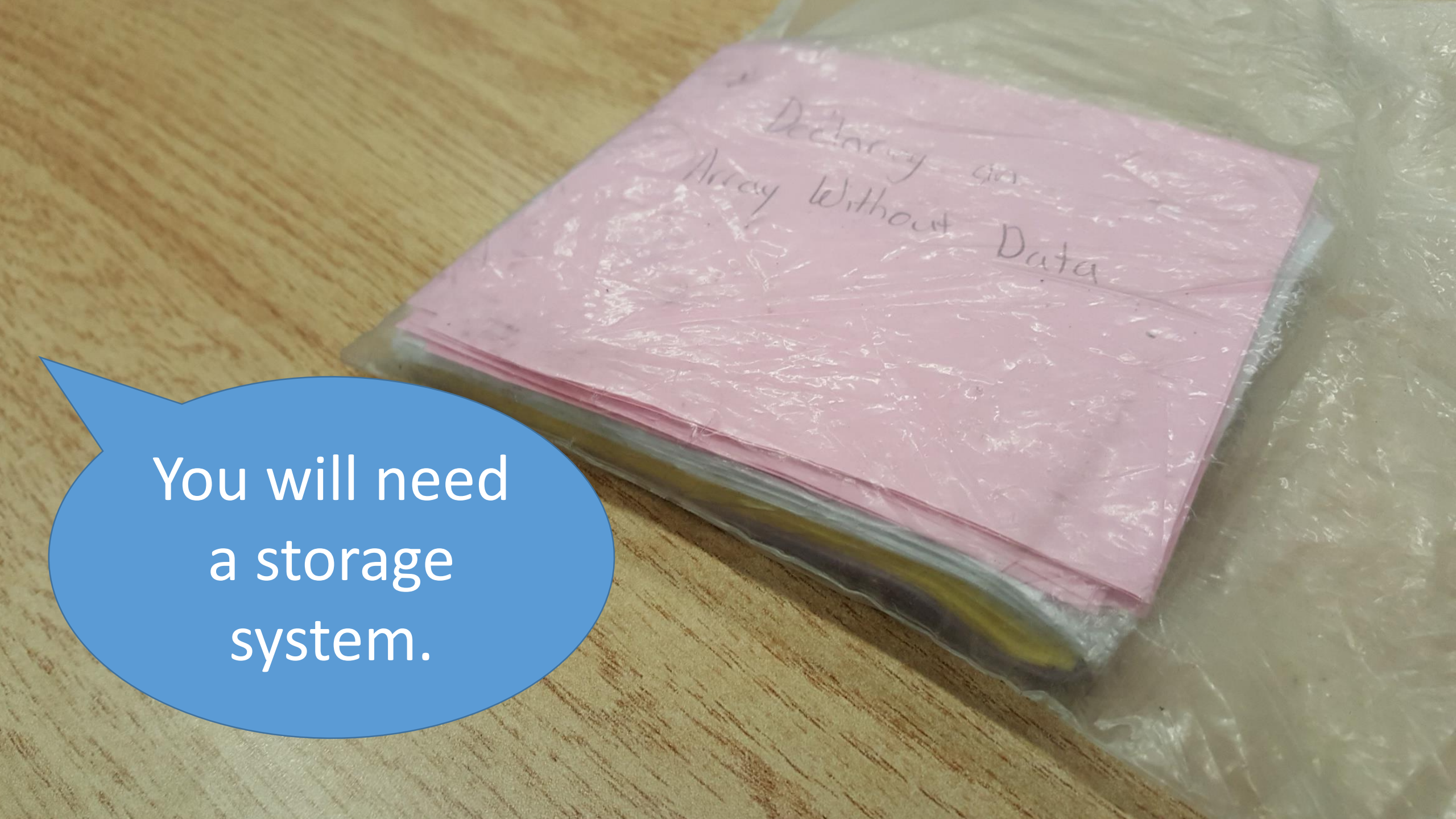
Question	Answer
What is a server?	- computer (normally big) on the internet that provides services - it might ~~host~~ host a webpage, or hold your social media, or hold your Alexa answers, or host Fortnite games
What connects client + servers?	- cables, normally fibreptic, connect computers + routers together - cables + routers together make up the network, are owned by ~~ISPs~~ ISPs (Bell, Rogers) - You pay an ISP to maintain cables + routes
What is a Packet?	- A piece of a message that travels between a client + server - It needs to and from addresses - It also has a sequence number for reassembly.
What is a Router?	- A computer on a network that directs messages to the right destination.

Fold


Declaring an array without data
int a[] = new int [4000];
type
Name
Makes an array
type
Size
JButton b[] = new JButton [26];
Take notes directly into DIY flash cards
Another technique

One colour per unit… or another system

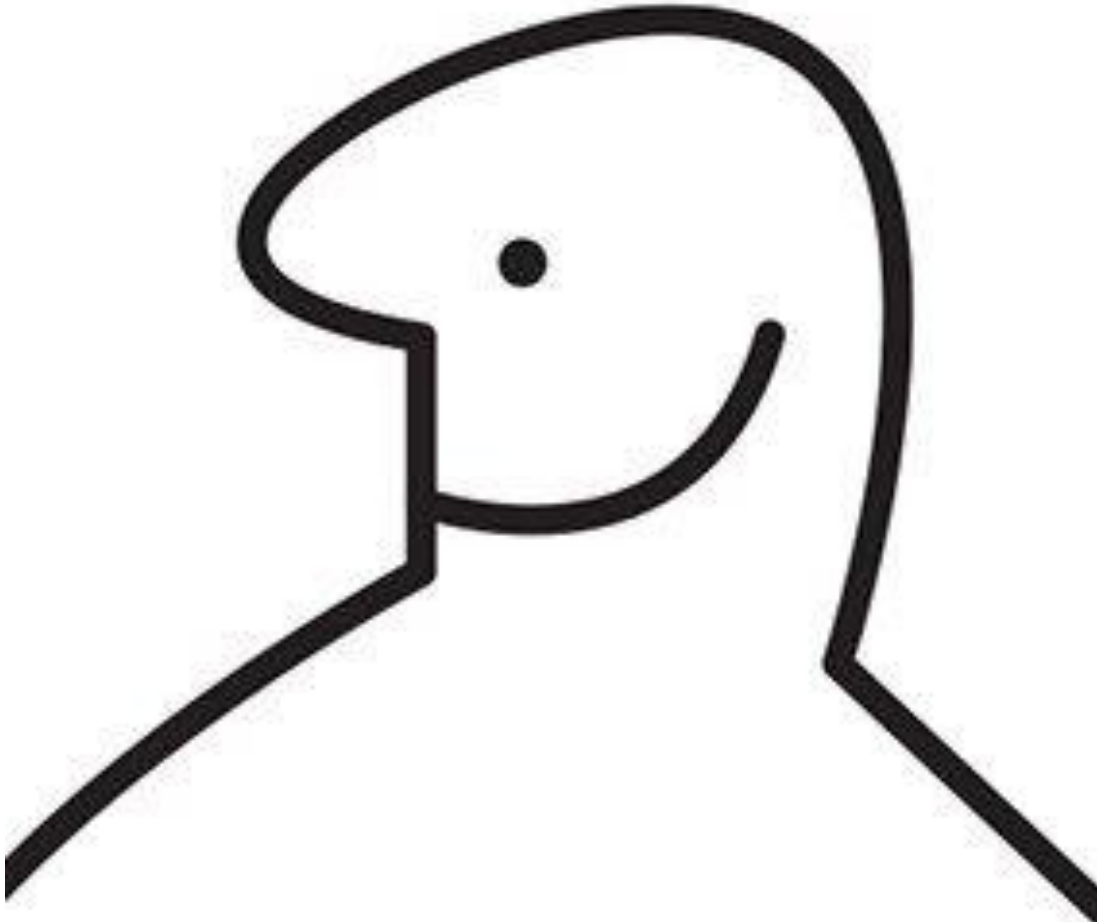



Array Without Data
You will need a storage system.

Practice

Info Processing

Memory Consolidation

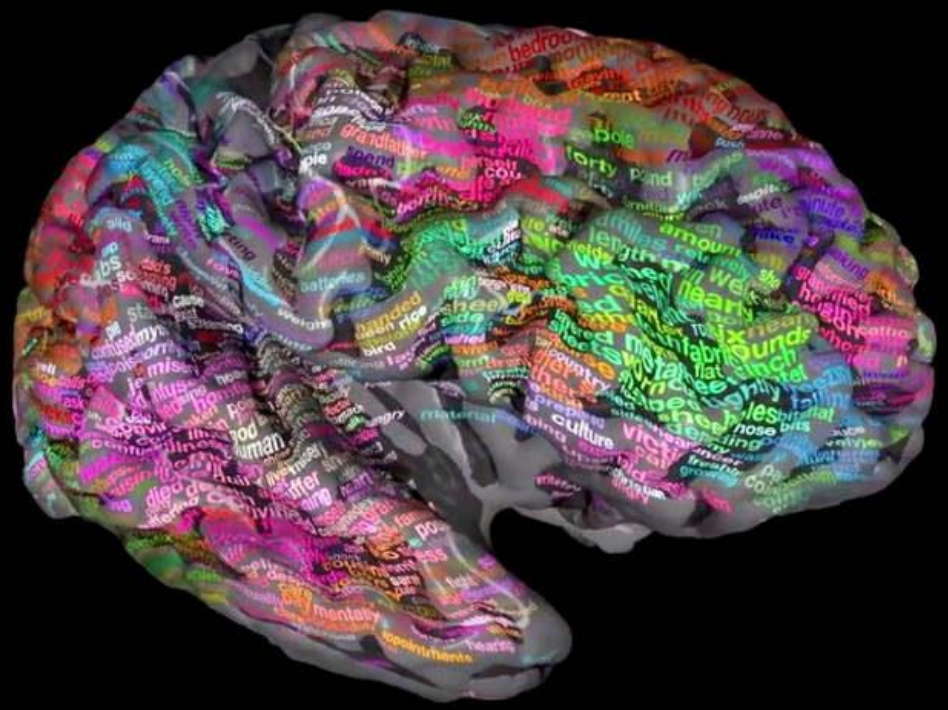


Critical in this process is asking how and why.

Why is this important?

Why does it work that why?

Why is this a stronger answer than that one?



You need to put information into your brain, then you need to consolidate it.

Generating Explanations: Presley et al

The strong man carried a shovel.
The toothless man wrote a check.

Group 1	Group 2	Group 3
No explanation	Given explanation	Generate explanation

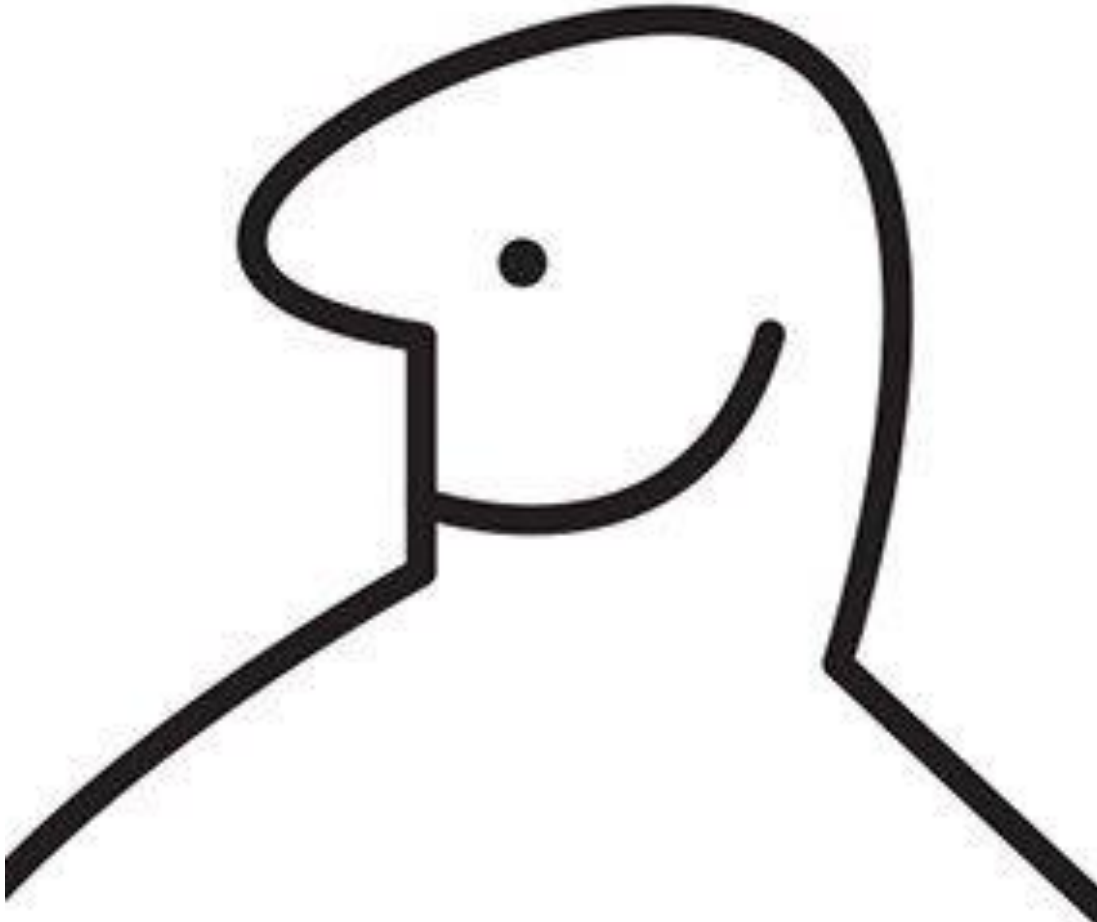
Generating Explanations: Presley et al

The group who generated their own explanations performed about twice as well as the other groups.

Visuals

Info Processing

Memory Consolidation



Human brain mapping and brain decoding. | Jack Gallant | TEDxSanFrancisco

Nishimoto, Vu, Naselaris, Benjamini, Yu & Gallant, Current Biology, 2011; Bilenko, Savage & Gallant, *unpublished*

11:46 / 17:26

Scroll for details

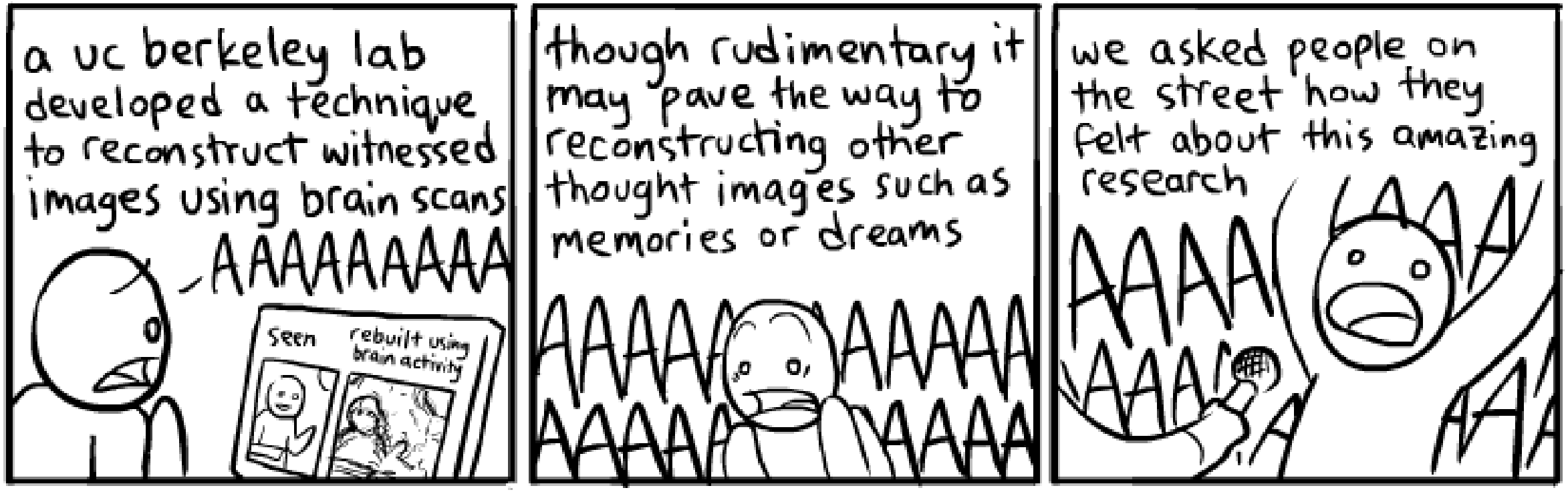
Presented clip

Clip reconstructed from brain activity

Presented clip

Clip reconstructed from brain activity

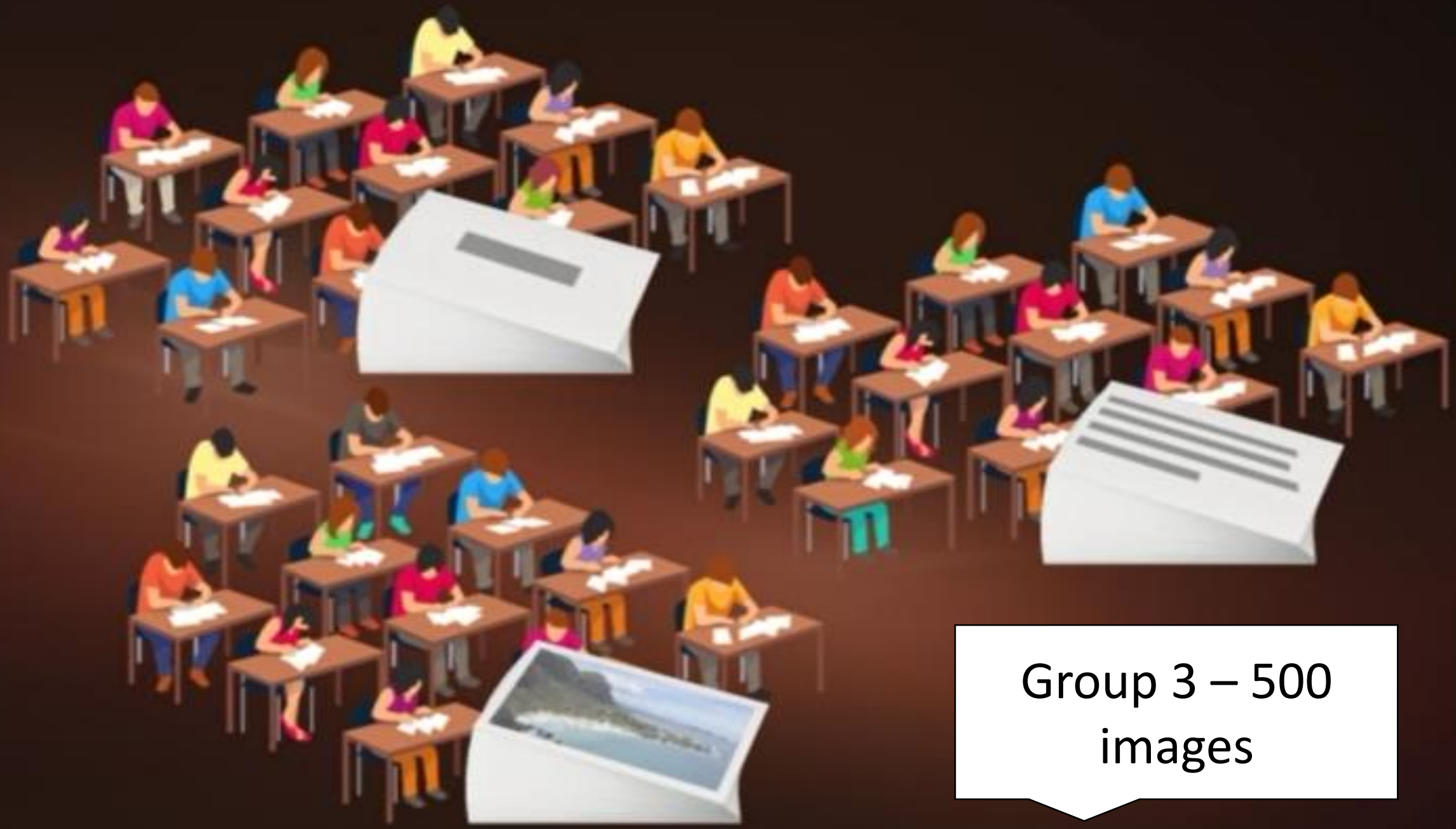
chainsawsuit.com



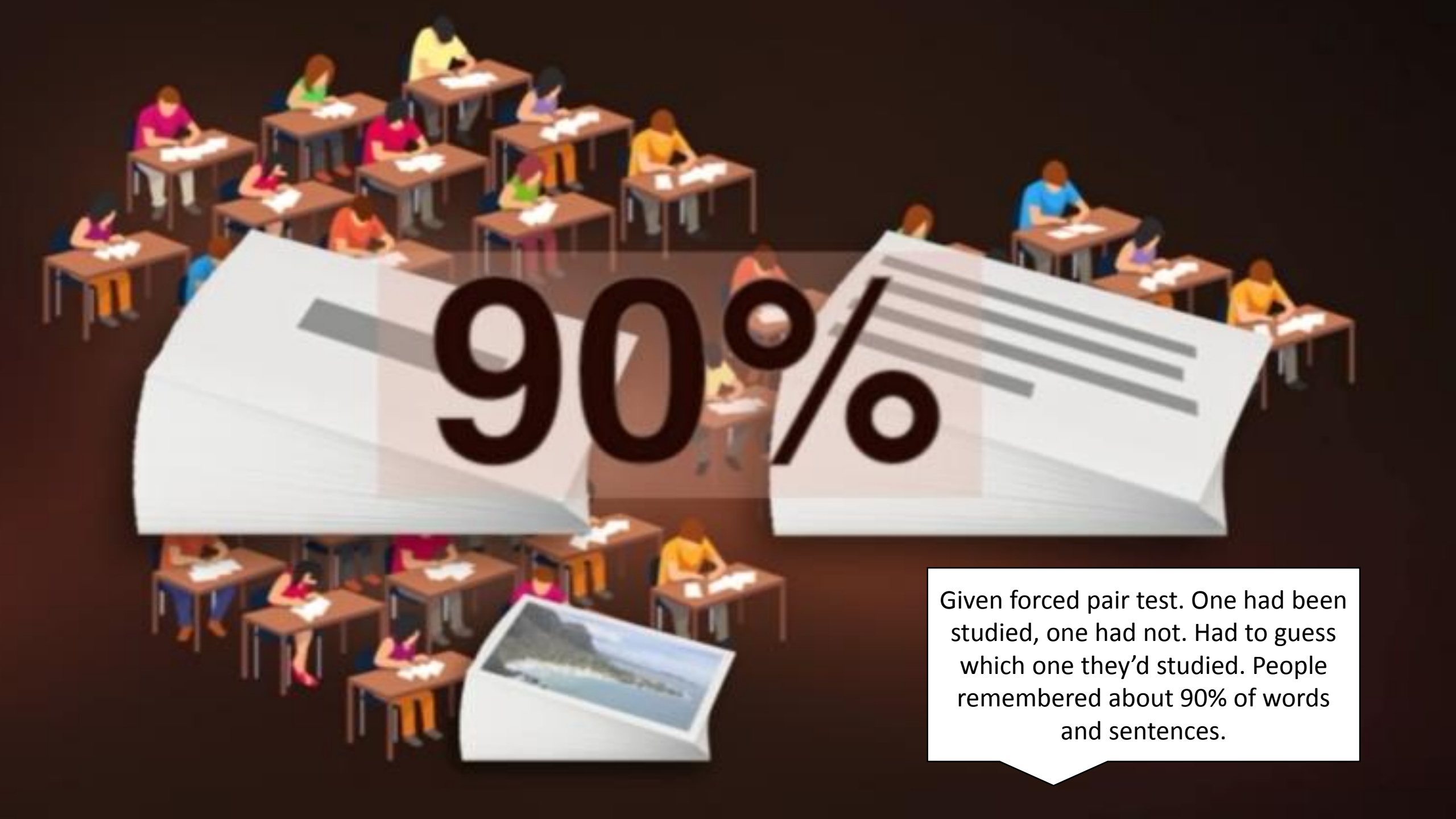
Roger Shepard’s study: 3 groups of students.

Group 1 – 500 index cards, one word on each. Look through them and learn how you wish.

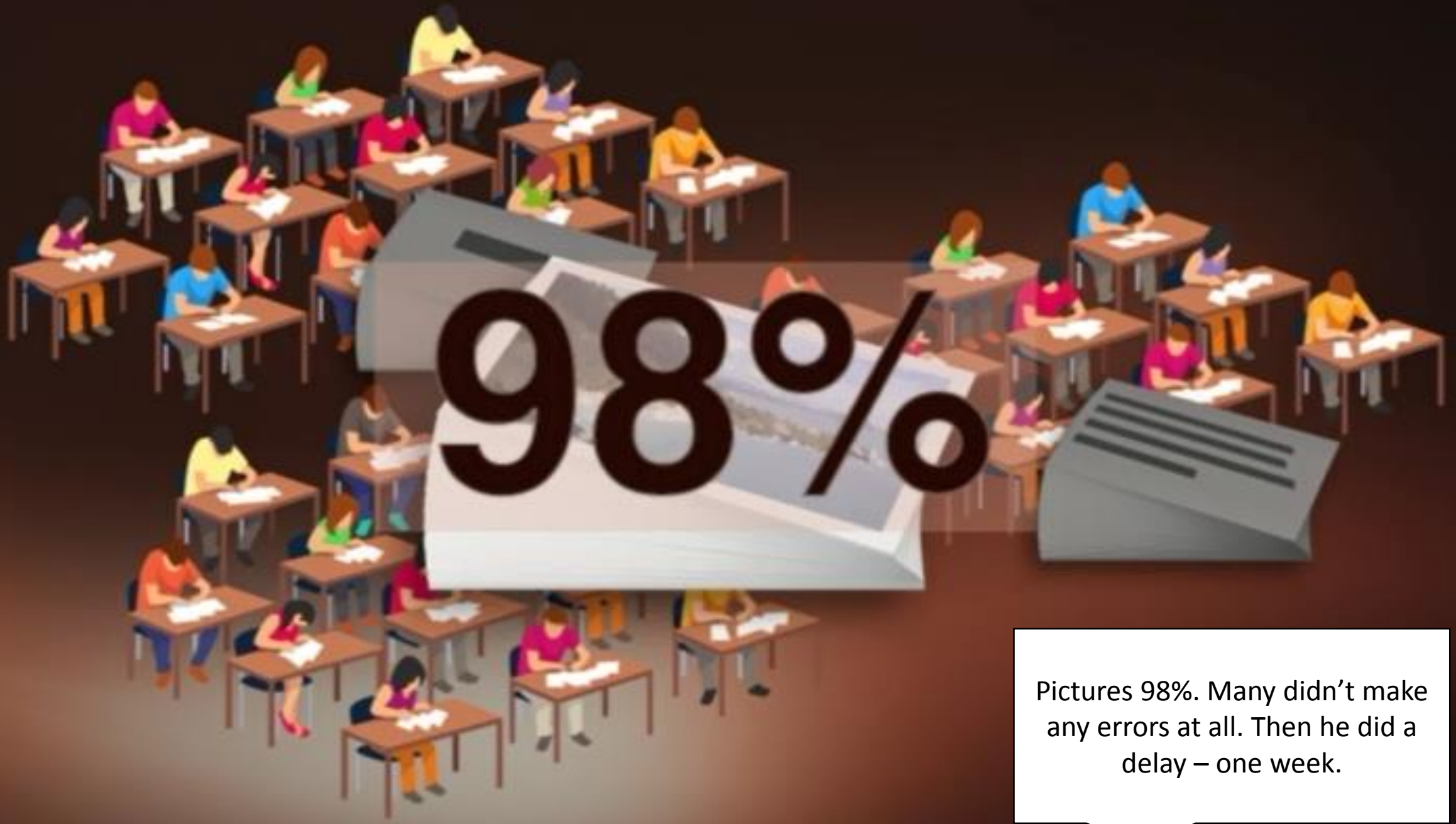
Group 2 – 500 sentences



Group 3 – 500 images



Given forced pair test. One had been studied, one had not. Had to guess which one they'd studied. People remembered about 90% of words and sentences.



Pictures 98%. Many didn’t make any errors at all. Then he did a delay – one week.

Shepard's Recognition Test Results

After one week – pictures = 90%

Memory for pictures you saw a week ago is the same as memory for sentences we just saw.

Leo Standing did a study where he showed people 10,000 pictures, each presented for 10 seconds.

Leo Standing did a study where he showed people 10,000 pictures, each presented for 10 seconds.

Took 3 days to show them the images.

Leo Standing did a study where he showed people 10,000 pictures, each presented for 10 seconds.

Took 3 days to show them the images.

Then asked forced-pair test.

Leo Standing did a study where he showed people 10,000 pictures, each presented for 10 seconds.

Took 3 days to show them the images.

Then asked forced-pair test.

They got 70% right.

Leo Standing did a study where he showed people 10,000 pictures, each presented for 10 seconds.

Took 3 days to show them the images.

Then asked forced-pair test.

They got 70% right.

No upper memory limit has been found for images.

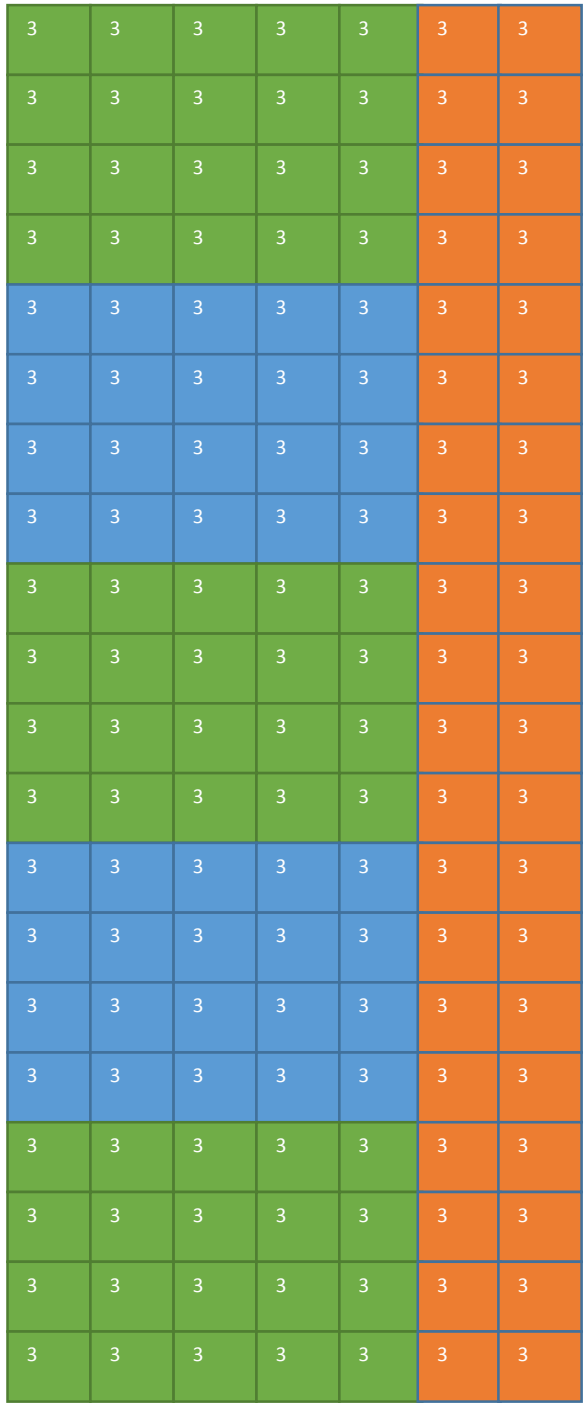
Sept

Oct

Nov

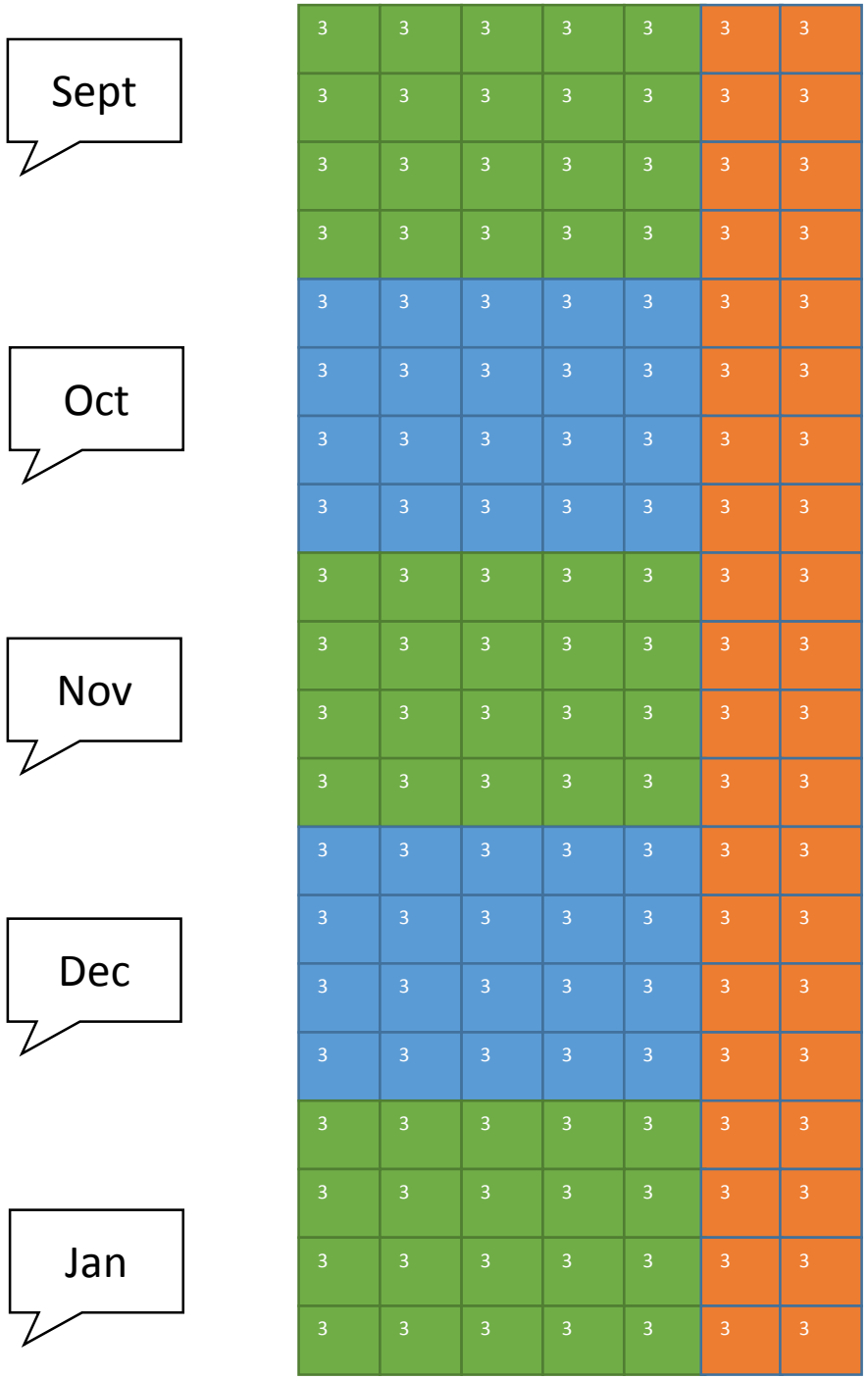
Dec

Jan



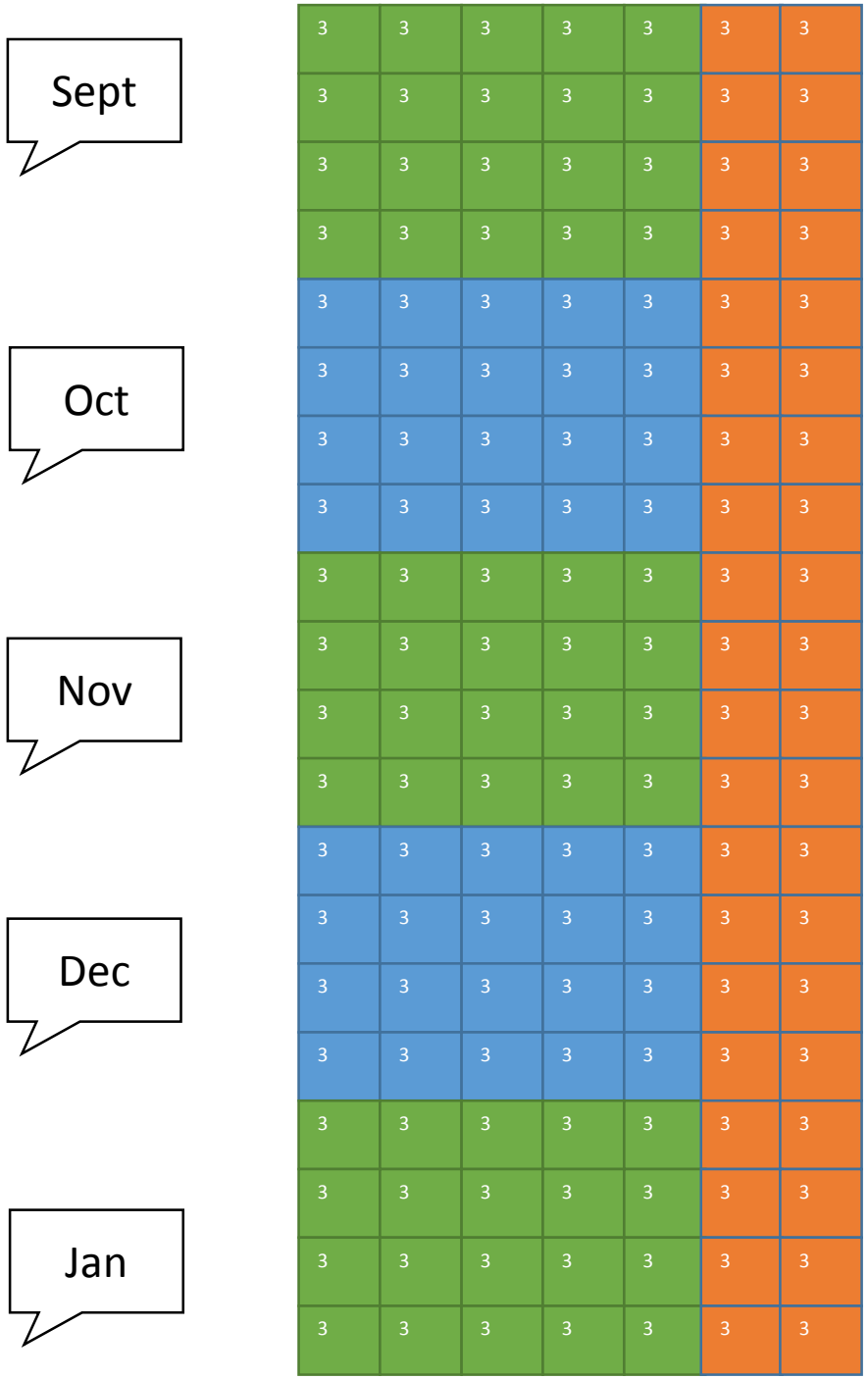
Organizing yourself

- Write in all assignments due dates at the beginning of the semester.
- Write midterms, major projects, finals in a bright colour.
- Regularly check the week and months ahead to make sure you are on track.
- YOU are in charge of breaking down your work and budging your time.
- On-line planners don’t allow you to see enough at a time to plan ahead.



Weekly assignments:

- Read the assignment EARLY.
- Read over your notes if you aren't regularly reviewing. Make sure your readings are done. Note things that match the assignment.
- Start work on them EARLY. Give yourself time to think about them. Give yourself time to go to office hours.
- Hand them in well ahead of time.



End of semester assignments:

- Budget yourself time EARLY in the semester to work on them.
- These require time to think about them.
- Make a plan and go and see your professors in Office Hours.
- At the end of the semester, you will be busy – a final edit is all you will have time for.
- No one else will be doing this – except for the top students.