

The Nine Principles of Immediately Effective™ Instruction

by Don Jones

ISBN xxx

copyright ©2013 Concentrated Technology, LLC
All Rights Reserved

www.ConcentratedTech.com

cover photo by Melony Marshall-Haynes
www.melonyhphotography.com

The Nine Principles of Immediately Effective Instruction

Contents

Foreword 4

You Can't Always Get What You Want. But If You Try
Sometimes... 8

Lather, Rinse, Repeat.
16

Don't Pre-Order - Deliver
25

Experience the Problem Before Offering the Solution
29

You Don't Know Their Life
36

Shut Up 43

Be Profound 46

And Now for the Top Ten
51

You've Got 30 Minutes to Make This Real
56

Afterword 62

The Nine Principles of Immediately Effective Instruction

Foreword

As I started writing this book, I realized that I had been teaching adults for almost 20 years. In that time, I've done about an even split of teaching my own material and teaching courseware written by others. I've also written more than 50 books, all of which were designed to teach some particular topic. Along the way, I started to become dissatisfied with the quality of that training. At the end of a class, I'd ask my students how they liked it, and they were always positive and complimentary (even in the formal, written, anonymous evaluations). But if I asked, "do you think you picked up anything useful?" they'd get a sort of dazed look. "I don't know." "Anything you can use when you get back to the office?" More dazed looks.

It was frustrating. As an instructor, the whole point of my existence was to teach these folks something they could *use*. My focus is on information technology training, helping IT professionals learn new products and technologies. After five days, they weren't sure what they'd retained. They weren't sure what would be useful back at the office.

I'd failed them, although based on their evaluations they'd had a great time maybe not learning much.

Many of my fellow instructors' job satisfaction began and ended with their class evaluations. Good evals, go home. I wasn't happy with that – I wanted my students to be *effective* after class, not dazed. So I started digging into instructional design for adult learners, to see what I was missing, and how I could do a better job.

Unfortunately, much of what's out there is extremely academic – to the point where I wonder if those instructional designers have ever actually taught a group of adult learners. There are a lot of arbitrary rules – no more than 5 topics in a lesson, no more than 4 lessons in a two-hour period. Communicate objectives up front,

and phrase them as gerunds. Don't call on people by name, because you'll embarrass them if they don't know the answer. Don't expect questions, because adults don't like to look like they don't know the answer in front of their colleagues. Seriously? This is instructional design and delivery advice? There were some nuggets of good information, but it tended to be really broad and hard to apply in a real classroom.

Too much of existing instructional design theory for adult learners assumes that you're teaching academic material – the proverbial reading, writing, and 'rithmetic. Vocational training, which is what *most* adult learners are interested in, seems relegated to a back burner. Even use the phrase “vocational training” with a trained instructional designer and you'll get a dismissive glare (I got a few). “Vocational training is for community colleges,” one told me. “I'm concerned with *real* education.” Well, excuse me. In my world, vocational training isn't just cooking and plumbing (which, by the way, are extremely valuable topics). Vocational training includes computer network design and operation, relational database management, enterprise systems configuration, elastic cloud infrastructure, and a lot more. Vocational training helps ensure that your banking information is kept private, that your medical records can be accessed quickly, and that your tax refund gets processed. In fact, nearly every well-publicized computer failure or breach can be traced directly to a *lack* of adequate, quality training. In my field, vocational training is what makes 100% of the Fortune 1,000 *run*. I'd say it's pretty darn important that my students learn new technologies and proper practices.

So I started experimenting. Taking those nuggets of instructional design wisdom that I could find, I started writing, re-writing, and revising my course outlines and delivery methods. I watched students like a hawk, and questioned them vigorously in class, figuring out exactly what was sticking with them and what wasn't. I paid

attention to body language, looking for the slightest hints of confusion or comprehension. I developed granular hands-on exercises that let me immediately test their uptake and retention. And I started to see some patterns.

This book is the result of almost ten years of study, experimentation, and exploration. I've codified my approach into nine distinct principles that leverage an adult's natural tendencies and patterns to help them learn new topics more efficiently. I call this approach *Immediately Effective*TM because it's designed to send someone back to work, or back into their lives, able to immediately use what they've learned. Students leave my classes excited and confident in their new skills. The full approach is currently used in a half-dozen instructor-led courses being delivered worldwide, and another half-dozen information technology books and instructional videos designed for self-paced study. Courses designed and written in this way can be taught by nearly any instructor who already understands the underlying topic, and they'll be markedly and measurably more successful. Instructors who know my approach, and who truly embrace it, can use it to make *any* class delivery more successful, by rearranging material and presenting it in accordance with these nine principles.

Nothing in my approach is magical, and frankly I don't think any of it is terribly revolutionary. Written down, it seems like common sense. The problem is that it *isn't* "common." We learn to teach by remembering how we were taught, often in primary or secondary education *where the rules are different*. Teach something to a child and you can break nearly all of my rules, and their spongy little brains will adjust. My principles simply reflect the realities of an older brain that seems a bit more resistant to learning – a resistance that, I discovered, actually comes from built-in survival instincts.

The Nine Principles of Immediately Effective Instruction

Any decent instructor, and any decent course author, wants their students to be successful. They want their students to *learn*, and to be able to *apply* what they've learned. With nine simple principles, *Immediately Effective* can make it happen.

You Can't Always Get What You Want. But If You Try Sometimes...

Teach students what they *need* to know, right then. Don't try to teach what you, as the instructor, *want* them to know.

It's easy to bring your own agenda into a class. After all, you're the instructor – you literally set the agenda. But the biggest area where I see instructors fail is in teaching what *they* want students to know, not what students *need* to know.

Most instructors and instructional designers are familiar with the idea of setting and communicating objectives. If your class is broken into modules, or your book is broken into chapters, you typically start out by stating your objectives. One of my publishers, Manning, has a standard where each chapter starts with a bullet list titled “In this Chapter.” Microsoft Official Curriculum modules start with a module overview that includes the module objectives. Those objectives are what you plan to teach.

Most instructional designers will also agree that objectives should be active, which is why you'll often see rules like “objectives must start with a gerund.” So, an objective like “SQL Server auditing” wouldn't be acceptable, but one like “Configuring SQL Server auditing” would be fine, because it starts with a gerund. That's supposed to make the objective seem more active. I'm not teaching you the feature, I'm teaching you how to *configure* the feature. That's a task you complete. It's a good objective.

Well, no. It isn't. Nobody wakes up in the morning with a hankering to do that. Configuring, and a huge number of other objectives I see in courses, is a means to an end.

It's what you do in order to achieve what you woke up hankering to do. It's a step. A process.

I dislike objectives in courses and in books. Each instructional unit – each chapter, or module, or lesson, or whatever you call it – should focus on one *need*. What is it that the student *needs to accomplish*? Title the instructional unit to clearly describe the need you're meeting, and everything else falls into place.

As an aside, instructional designers do have language for this concept. In their terms, I'm setting the chapter title as a *terminal objective*, and I might use one or more *enabling objectives* to get there. I still think that structure leaves a lot of room for an instructor to get sidetracked, though. I'd rather state what it is I'm going to teach you, and then ensure that every single word out of my mouth, or on the page, leads directly to that goal.

I might title a chapter something like, "Capturing User Activity by Using SQL Server Auditing." That's a statement of something a student might truly want and need to know – how to capture user activity. It's a job task. It's a human-level task, something the boss might tell you to do. It scopes the chapter to a particulate technical feature, but the statement is clear about the real-world ability you're going to be learning.

You then have to be really, really careful about what you teach to lead to that goal. The line between what the instructor *wants* to teach and what the student *needs* to know can be really fine.

Let's take a non-technical example. I'm teaching a class, and we've gotten to a module I've named "Using a Car to Drive to the Supermarket." That's a pretty clear title, although instructionally I may have several sub-tasks I need to teach: starting the car, putting it into reverse, and so on. Those are things the student *needs* to know in order to achieve the goal stated in the module's name.

The Nine Principles of Immediately Effective Instruction

You can't get to the market without starting the car and learning to steer it in different directions.

As an instructor, I *want* my students to be well-informed. I *want* them to be smart. I also *want* to impress them with my own knowledge, so that they think I'm smart and give me a good evaluation at the end of class. And so I decide that they *need* to know how the brakes on the car work. You depress the brake pedal, an electrical signal goes to the master brake cylinder, and that cylinder compresses and pushed brake fluid to the brakes.

Wrong.

Students *do not need to know that*. Knowing that information does not in any way help them achieve the goal stated in the name of the module. It might help them with *other* goals, like troubleshooting or fixing the brakes, but that wasn't the deal we made with them when we named the module. What they *need* to know is that you stomp on the brake pedal and the car stops. That's it.

Here's why: human brains can only absorb a certain amount of information in a given period of time. After that, those brains literally need to sleep on it, allowing the brain's natural background processes to organize the new data, make relations and connections to existing data, and get everything neatly filed in the neural data store we call "memory." Feed someone too much information at once, and the brain goes into an automatic triage mode: *I can't possibly remember all of this. Let's see if I can figure out what's important, and I'll just store that*. The problem is that this triage mode isn't under conscious control, so the student doesn't get to decide what's extraneous and what's vital. The brain makes that decision almost instantly, based on its ability to connect new information with existing information. In other words, the student's brain might decide that the whole tangent on master brake cylinders was the important bit,

and not actually capture the critical fact that *stomping on the brake stops the car*.

So you *must not* introduce extraneous information. Teaching isn't about what you *want* someone to learn, it's about what they *need* to learn in order to absorb the skills that you promised to deliver when you sold them the class, the book, or whatever.

I teach a lot of Microsoft Official Curriculum courses. Microsoft has a lot of challenges when it comes to courseware, not the least of which is the highly variable quality of their certified trainers. In one instance, I was teaching a class that had previously been taught by another guy at the same facility. That man knew this material cold, and a lot more besides. He knew the product inside-out, upside-down, and sideways. He took every minute of the recommended delivery time, and then some, running students from 8:00am to 5:00pm every day, and even shortening lunches sometimes to pull in extra time.

When I taught the class, we went from 9:00am to 3:00pm, had a full hour for lunch, and my students were measurably better at retaining and applying the skills that the course had promised to teach.

The difference? I threw in no tangents. I added no extra information. I didn't try to prove I was smarter than anybody, and I taught the class with no regards for what I *wanted* the students to learn. I took the course's top-level outline – what students had seen when they paid for the class – as what they *needed* to learn, and I taught nothing that didn't directly lead to those needs. The fact that I had a shorter day kept my students fresher, meaning they didn't hit the “fatigue wall” in the afternoon and stop learning outright. Their brains didn't have to go into triage mode and start indiscriminately dropping new information. They walked away feeling more confident that they'd captured everything I'd

taught, because I hadn't overloaded them with a firehose of information.

When I first started writing, I wrote long-form IT books. I had no real page count limit, and these things would be 600, 800, or even 1,000 pages long. Then I managed to land a monthly column in a trade magazine, where I'd be allotted just 800 *words* each month. My editor at the time, Keith Ward, told me that writing an 800-word column would make me a better writer pretty quickly, because I'd learn the precious value of every single word. He was correct. These days I have trouble writing books of more than 300 or 400 pages, because I've learned to be more concise, and to stay on-topic. My books are better not because they're shorter, but because they do a better job of teaching – in large part because they're more *concise*. That's the real purpose of the principle I'm writing about, here: be *concise*.

There's another lesson here, which is the implicit contract created between an instructor or author and his students or readers. When you name a chapter, or provide a list of objectives, or whatever, you're promising to deliver just that. If students walk away feeling like they haven't learned that, then they feel cheated – even if the “failure” was you pushing so much information at them that their brains couldn't cope. In other words, if you *over-teach*, you may well fail to deliver on that implicit contract.

The implicit contract is also important in preparing the brain to learn. *These are the things you will know*, you're telling each student. *This is what I will teach you*. That statement can actually help position their brains to learn those things – provided you stick to the contract and don't wander off-message.

This brain behavior is built-in, and you see it used in all kinds of different situations, often to great effect. Magicians, for example, rely heavily on the fact that our brains will look for whatever we're told to look for – and

they use that to distract us while they're palming the card or getting a dove from their coat pocket. The minute our brains are told what to expect, we start subconsciously looking for that. So it's important that, once you've created the contract on what a module or chapter will cover, you *immediately* start delivering on that contract. Every concept you introduce must be *explicitly* connected to what you said you'd teach.

"I'm going to teach you how to stop a car. Before I get into the actual procedure, I need to talk a bit about momentum, because the concept of momentum is going to affect the procedure." In that sentence, you're acknowledging that you're about to seemingly go off-topic, but you're immediately providing a connection back to the promised topic. "The more momentum you have, the harder you will have to brake in order to stop the car at the same stopping point." Now you've firmly connected the procedure to your tangent. *More momentum, more braking*, students' brains will think. *Got it.* Now you can go into more detail about what momentum is – but keep it relevant. Don't discuss Isaac Newton – he may have discovered momentum, but he does nothing to further the contracted goal of stopping the car. Keep the momentum diversion as brief as possible: "The faster your car is going, the more momentum you build up. That momentum must be reduced to zero in order to stop the car. The brakes can reduce momentum, but the more you've built up, the faster you'll have to reduce it in order to stop at a given point."

So what about student curiosity? What about the student who is really clever, totally grasps what you're teaching, and wants to know some of the underlying details?

Curiosity is good, and should be encouraged – but not at the expense of the less-curious in class. Instead of going off on a more-detail tangent just to prove that you, as the instructor, really do know this stuff, encourage

independent exploration. Build “further reading” references into your instructional materials, and encourage curious students to explore. By doing so, you’ll achieve two goals. First, curious people usually enjoy independent exploration, and by encouraging it you’re encouraging a smarter person. Independent exploration engages a number of higher brain functions, and often helps a student’s brain better cement new knowledge. Second, you’re keeping the rest of the class on-track. Unless you and the curious student can leave the room – something not possible in a book, by the way – you’re putting your tangent in front of everyone. Folks will listen to even a “private” in-class conversation, to be sure they’re not missing something, and now you’re bombarding everyone with extra information. Brain goes into triage mode, class success level goes down.

In books, tangents are often styled as sidebars. They’re often set aside visually in some way, perhaps with a gray background or in a box. Readers are often told to ignore them. The *Dummies* books (and I did write one of those) label them as something like, “Dumb Stuff You Don’t Need to Know.” Problem is, you can’t *stop* someone from reading, and they’ll often do so because they fear “missing” something. If they don’t need to know it, don’t put it in the book. Period.

Many of the classes I teach include remote students. Early on, I would shut off my microphone during labs, so that I could discuss issues with each student individually. I relied on the full-room microphone in the ceiling to let remote students know I was still in the room. I’d get into “expanded” discussions with a curious student, and the remotes would invariably, to a person, complain that they couldn’t hear me. I wasn’t talking to them, but they were worried they were missing something. Once I realize that I was breaking my own rule of not introducing tangents, I stopped doing the side discussions. Maybe on breaks, or during lunch, outside

the classroom, where it's more clear that the conversation is private, but never in class.

Mick was right: you can't always get what you want. But if you try sometimes, you might find you get what you – or rather, your students – need. When you begin a new unit of instruction, clearly state what students will learn. Do so in terms that are meaningful to the student's life – state goals that represent real-world tasks and abilities. Ensure that every word out of your mouth, or every word written on the page, ties directly to that goal. Teach *nothing* that does not apply directly to that goal. Teach what they *need*, and not what you *want* them to know.

IMMEDIATELY EFFECTIVE™

Students have a limited ability to learn in a fixed period of time. Make sure you're focusing on the most crucial items, and teach *only* those. You'll help students be more effective when you take on the responsibility of triaging the information.

Lather, Rinse, Repeat.

Repetition is important... but only when it's the *student* that's doing the repetition.

Anyone who's been to a Disney park can tell you how well repetition works. *It's a small world after all, it's a small world after all...*

Instructors and instructional designers have long known about the value of repetition. Keep telling someone something and eventually they'll remember it. The concept is put to good use with primary education in K-12, and even in college: students take notes (repetition), do homework (repetition), and recite facts on demand (repetition, repetition, repetition).

Unfortunately, most of those techniques aren't available to adult educators. Adult learners who are aggressive note-takers aren't engaging in repetition, they're just missing the lecture or demonstration. The goal becomes copying everything down, not absorbing it, and their notes lack the nuances and subtleties of the original delivery, making the notes a poor way to actually learn the material. Adult learners simply won't do homework, so that's out. We also don't call on individual learners in class for fear of embarrassing them, so verbal quizzing is limited to throwing out questions to the group and waiting for the "smart guy" in class to answer.

Adult educators still recognize the value of repetition, so lacking these other mechanisms they tend to just say the same thing over and over. It's all they've got. So they end up droning, *repeating* themselves ad nauseum and losing the attention of their students.

All of which misses the whole point of this chapter.

The famous phrase on the bottle of shampoo says, “lather, rinse, repeat.” Of those three tasks, which are the most important?

Lather and rinse. That’s what actually washes your hair. Do you really need to repeat? Probably not. I never do, and my hair stays pretty clean.

That is the whole point of this chapter. Yes, repetition can be important – but what’s more important is *physically taking action*. Let me give you an example.

I often teach technical courses where the courseware has the instructor perform some kind of product demonstration. There’s usually a mix of lecture and demonstration, in which the lecture explains why or how something works, and then the demonstration shows how it works. That’s a form of lightweight repetition, and the general consensus is that it works well to reinforce key concepts and show the student how to do something. After a while of doing that, students go into a hands-on lab where they often perform the same tasks that were just demonstrated – more repetition.

What I found through some experimentation is that the demonstrations are almost completely ineffective when used in that manner. Adults have to do some odd context-switching when you move from talking to them to showing them something. The amount of time needed for that context-switch isn’t long, but it isn’t zero, and they end up not absorbing most of the demonstration, as well as missing a few key points when you switch back to lecture. It’s like putting someone in the passenger seat of your car, driving them to a location they’ve never been before, and explaining something to them the whole way. They’re not going to learn the route you took and be able to get to that destination on their own.

As an experiment, I started skipping demos where I knew the same tasks were in the upcoming lab. I stuck with the lecture, and then dumped students,

“unprepared” into the guided lab. They were much more successful. They stumbled a bit here and there, but frankly they’d stumbled the same way when I’d just demonstrated how to do whatever the task was. The demonstration added nothing, and dropping it had zero impact apart from saving us time and keeping students’ minds more on-track. After the lab, students were more confident of performing their tasks – I could give them an ad-hoc exercise related to the lab, and they’d be able to complete it more quickly and more surely.

Doing something is how adults learn.

There’s another technology I’m well-known for, and I’m probably one of the more experienced instructors for that technology. One of my personal quirks is that, when I teach it, I don’t use slide decks or perform the normal lecture-style presentation. My “lecture” is almost 100% demonstration, coupled with me waving my hands around a lot while talking. Students like the approach, because it’s more interactive. Also because, quite honestly, staring at slides is boring. But I discovered that my approach had some flaws. Students *don’t learn from demonstration*. There’s too many brain components working at the same time for actual comprehension to kick in for most folks. What’s happening on the screen up front is completely detached from students’ existence. Their eyeballs and brain have to pick out the bits on the screen that are moving, relate that to some hazy goal they’ve been told, all while memorizing task sequences that don’t yet have any context for them.

Cooking shows do this all the time. You watch the TV cook prepare some beautiful dish, all the while waxing poetic about the quality of the butter, that time in Napa they had this same dish in a wonderful restaurant, and how you have to be careful not to use too much oil or you’ll burn the pan. The facts are flying fast and furious, and they don’t always precisely match what’s presented to your visual cortex. Nobody learns to cook by

watching TV. At best, they get a recipe idea, which they then have to look upon the channel's Website to actually try.

Outright demonstration isn't always an effective learning tool, even when it's not being used simply for repetition. Students have to *do something*, themselves, before they'll actually grasp something.

I've since modified my approach. Knowing that the lab – the student's hands-on “lather, rinse” activity – is the important bit, I've shortened both lecture and demonstration sequences. I talk and/or demo for as little as possible, and focus on giving students the information they can't self-discover, such as background concepts. I teach *one* thing, and then try to get them to a lab where they can practice that *one* thing as quickly as possible.

What about repetition?

It's in the labs. I design my hands-on activities so that students not only practice what I *just* taught them, but also re-visit tasks that they learned earlier in the class. I don't re-teach those things, but I make students re-practice them frequently. By the end of the class, the things they've repeated the most, as you might expect, “stick” really well.

And therein lies the instructional design challenge. There are really two principles here, neither of which seems very revolutionary, but both of which can be pretty hard to actually achieve:

- Your class' most important skills need to be introduced as soon as possible, so that students will have the most amount of repetition with them throughout the class. Everything you teach one day 1 gets repeated every other day.
- Your class has to use a very carefully designed outline, so that you're constantly building.

Chapter 1 has to teach something that will be reused in Chapters 2, 3, 4, and so on. That's where the repetition comes from.

These two simple facts drive one of the most important instructional design activities in the world – one that a lot of course authors skip or don't do properly: *triage*.

Look, you can't teach someone everything about a complex topic in a single class. You just can't. You *have* to pick and choose. Let's say you're teaching them about driving a car. You decide, early on, to teach them how to change the oil in the car. How often are you going to repeat that in subsequent lessons? Probably never – which means you either shouldn't be teaching it, or you should be teaching it last. If it doesn't naturally lend itself to repetition throughout the class, then it isn't a core concept, and it either belongs at the end or it doesn't belong in the class at all.

These two principles really do force you, as a designer, to identify *core* content. You can only teach an adult about 3-4 new things an hour before their brains stop absorbing. You know the course's earliest hours have to start with things that can be repeated in every upcoming hour. You know you can't teach something that depends upon something else that you *haven't* yet taught. The course outline almost writes itself.

You *will* wind up discarding material – not teaching them at all – that you thought were absolutely crucial. That's a natural side effect of properly applying this chapter's principle. If something doesn't bear constant repeating, and isn't something students can actually perform hands-on, right then, then the material gets shifted later and later in the course. It's entirely probably some material will get shifted right out of the course altogether... and that means the material wasn't so crucial, after all.

Not coincidentally, this approach helps you better define class prerequisites. As you start to build your outline, you may find that you want to shift *early* material “forward.” In other words, you’ll identify key concepts. “I can’t teach them how to boil water until I teach them how to turn on the stove. But I don’t want to go that basic – I want to shift ‘turn on the stove’ to a prerequisite. They have to show up knowing that skill.” That’s a legitimate design decision – so long as it’s documented and communicated to students. It’s something that the instructor should review right at the start of class, so unqualified students can at least *know* they’re unqualified. Whether they choose to address the issue is their own problem.

Let’s look at a straightforward example of this entire approach, with the goal of teaching someone to put up the framing for a house. We’ll presume that the prospective student isn’t *designing* the framing, but that they’re simply being asked to physically implement someone else’s design. They start with a pile of lumber, boxes of nails, and some basic tools. I’ve actually seen short courses on this topic, and they’ll often start like this:

- The purpose of framing
- Structural loads and framing standards
- Building a basic wall
- Raising and bracing the wall
- Connecting walls
- Framing doors and windows

Those first two bullets shouldn’t be there. I *hate* teaching abstract concepts on their own, because it forces students to queue up material in their brains for later application, something the adult mind does not like to do. Ask yourself, “how many times will students have to repeat an exercise on the purpose of framing?” The answer is

zero, and that means the item should either be moved down the list, or off the list.

This outline misses some basic points. What's the proper way to hammer nails into the wood? How should the wood be cut? *Those* are key skills that *will* be repeated in every upcoming lesson, and those skills should either be taught first, or clearly listed as prerequisites.

The rest of the outline may be in pretty good shape. "Building a basic wall" seems to be something you'd repeat: you'll build one before raising one, you'll build one before connecting one, and you'll build one as part of framing in a window or door. Raising and bracing seems like something you'd repeat a lot, too, so that earns a spot early on in the class.

This example brings up an important practical point, which is the balance between the need for hands-on repetition and the logistical limitations on how long you've got to teach. In an ideal world, you might structure your class like this:

- Nailing and Sawing
- Building a basic wall
 - o Sawing the wood
 - o Nailing the pieces together
- Raising and bracing the wall
 - o Sawing the wood
 - o Nailing the pieces together
 - o Raising the wall
 - o Bracing the wall
- Connecting walls
 - o Sawing the wood
 - o Nailing the pieces together

The Nine Principles of Immediately Effective Instruction

- o Raising the wall
- o Bracing the wall
- o Connecting the wall

In other words, each new section repeats *everything* that has come before. That's a learning ideal, but it's rarely practical. Due purely to time considerations, you'll have to forego a certain amount of hands-on repetition. Some courses do this by continually building on previous work: you build the basic wall, then you raise and brace that same wall, then you connect that same wall to an existing wall, and so on. That's *zero* repetition, and while you might have to go that route in order to teach the necessary objectives in the time allowed, it's far from ideal.

The ideal is probably somewhere in the middle. Build a wall. Raise and brace it. Build a *new* wall, raise and brace it, and connect it to the first one. Core skills, taught early, are getting repetition with that approach – you're building *two* walls. But you're not starting from scratch every single time.

The overall theme of this chapter is to teach as little as possible before letting students practice something themselves. Work in small steps, through a carefully planned progression that permits explicit and implicit repetition of core skills. Give students pre-constructed “starting points” only as a way of meeting time restrictions, but don't completely eliminate hands-on repetition. The skills you teach earliest, and repeat the most often, are the ones students will truly take home with them, so strategize carefully and triage the material as you must.

And never forget that it's the lathering and rinsing that washes your hair.

IMMEDIATELY EFFECTIVE™

Give students plenty of opportunity to encounter and re-encounter the most important items you are teaching. Repetition is important, but students must actively repeat, not simply be repetitively lectured.

Don't Pre-Order - Deliver

<p>Foreshadowing is fine for Shakespeare. You're not Shakespeare.</p>

My single biggest pet peeve is a class where the instructor mentions something, but doesn't immediately cover it – “we'll cover that later.”

Now, while you're learning to care for camels, you should also know about their one magical super-power, that can make you rich beyond your wildest dreams. But we'll cover that later.

Every single time an instructor does that, at least half the class is thinking, *no, cover it now!!* I'm actually okay with “we'll cover it later” in response to a question. If a student asks something, and you know you'll get to it, then by all means let them know you'll get to it. You'll eliminate a possible tangent, satisfy their immediate curiosity, and if you're a really good instructor you'll acknowledge them again when you actually do get to whatever it was.

I'm not okay with “we'll cover it later” being brought out in any other way. Sequence your material so that you don't have dependencies on future material. That is, until it's time to talk about something all the way through, *don't talk about it at all.*

Foreshadowing creates a cognitive “to do” list in some students' brains. They start trying to create connections to material that you haven't even covered yet, and they start building this list of expectations for what's coming up. None of that serves any useful learning purpose. If it's coming up, students don't need to *know* that. They'll be there when it comes up, right?

Worse, foreshadowing immediately sidetracks the brain from whatever you *are* covering right at that moment.

Let's explore this from an instructional design perspective. The goal of instructional design is to create learning materials that do the best possible job of teaching some topic. In order to do a good job at instructional design, you have to keep in mind how your learners' brains are reacting to what you're doing to them. Humans - especially adults - have funny brain stuff going on all the time, and a ton of instructional designers just completely ignore it.

For example, let's take the concept of stating a module's (or chapter's) objectives up front. I'm not a fan. Yes, you absolutely need to give your learner some idea of what you're about to cover, but as I wrote in this book's first chapter, your *title* should do that. "Riding a Horse Without Falling Off" is a pretty clear explanation of what a chapter is going to cover. Immediately following that with a list of objectives?

- Mounting the horse
- Sitting in the saddle
- Holding the reins
- Walking the horse
- Moving into a trot

Blah, blah blah. *The learner doesn't care*. Those are what instructional designers call *enabling objectives*, meaning they're things you have to cover in order to deliver on the promise of the chapter title. Ultimately, students only care if you deliver on that promise - they don't necessarily need an outline of how you're going to get there.

Here's a perfect, real-world example that any traveler has experienced. "Hello, this is Captain Dave. Welcome to flight 1234, headed for Philadelphia. On the way, we'll

be passing over the Rocky Mountains, and you'll be able to get a view of Denver off the left side of the plane. We'll be turning South a bit to avoid some weather patterns over the Central Plains, and turning North again around Arkansas."

Seriously? Am I supposed to remember that? Should I have brought a map? Look, Dave, I paid you to get me to Philly. How you go about it is your problem, and I don't need a rundown.

The point is that presenting that huge list of objectives is the same thing. It isn't actionable information - the student can't *do* anything with that list right then. But adult brains will grab on to that list because we *love* lists. Checklists especially. So now the brain will start trying to make sure we hit all the items on the list. It isn't setting itself up to be *receptive* to those items, it's just *looking* for them. Instead of learning, it's trying to complete a collection.

Then you hit it with the "we'll cover that later." "Oh, and along the way, folks, we're going to be testing out a new APU in the airplane. That's an important piece of equipment, but I'll tell you about it later." *Crap*. Now my brain wants to hang on to that, and I'm going to fret about what the stupid APU is, and why can't you either just tell me *now* or shut up about it until the time comes?

That's the key lesson of this chapter: If you're not ready to talk about it *right now*, then it isn't important *right now*, so just keep it a secret so that I can focus on whatever *is* important right now. My brain is working pretty hard already in your class (or book, or whatever), so let's just avoid giving it a bunch of future expectations, okay?

Following this rule requires the most careful possible sequencing of material in a class or book. So much so, that I routinely refuse to deliver pieces of a work until the whole thing is done. I'll be in Chapter 5, ready to

write about "x," when I realize I have to cover "v" and "w" first, and that I haven't done so. So I have to restructure, rewrite a bit, and move on. "Ugh, if I bring *this* up now, I'm going to have to cover it very lightly, and then touch on it in more depth later. That being the case, maybe I should just not cover it at all until I can do it all the way. Later." For example, "steering a car is really easy, but it's similar in many ways to steering a motorcycle. We haven't talked about motorcycles yet, but for now you just need to know that they have two wheels. We'll get to the rest later." This is one very common use of foreshadowing - pulling in some example, but not wanting to get off on a tangent and fully discuss it right then. Use a different example. *Don't* set up the "to-do" item that foreshadowing creates.

Now, that said, you *can* cover something lightly, and then circle back to it for more complete coverage later. Just don't tell students you're going to circle back later. Just briefly cover whatever you're going to cover, and leave it at that. Students don't *need* to know you'll be covering the same topic in more depth later - they'll know when you get there. Again, if a student *asks*, it's okay to put them off - "yes, we'll cover that in a bit" - but you don't need to make a big deal of it.

So: sequencing. It's the magic that makes a good class into an awesome class. Always present material so one topic leads naturally into the next, without having to "refer ahead."

IMMEDIATELY EFFECTIVE™

Minimize distractions by not foreshadowing information. Sequence material so that students only have to digest a single stream of information, rather than building a "buffer" of "what's coming up next."

Experience the Problem Before Offering the Solution

Until you care, you just don't care.

Humans are often described as tool-using animals. That description, however, missed the point. Sure, we make and use tools all the time - but why? We typically make tools to enhance our own physical abilities, or to accomplish things we can't physically accomplish ourselves. In other words, we make tools to *solve problems*. The hammer wasn't invented until someone got tired of banging rocks on things. In other words, we don't just go around inventing things for no purpose - we invent things to solve problems.

It's stunning how often this fact gets missed in learning materials, but there's a reason: kids are sponges.

This is going to take a minute to explain, so bear with me.

Kids tend to learn without needing reasons. In other words, you can teach them abstract concepts, not really explain *why* they need to know that thing, and kids' spongy little brains will soak it right up. You absolutely *shouldn't* teach that way, but you can get away with it. Elementary school math is a perfect example, as was the horrible - and now largely discontinued - practice of making kids diagram sentences. We teach kids to add one and one, but we never really both telling them why. In other words, we give them the solution in advance of the problem. Sometimes, it's just easier to do so - can you imagine, for example, trying to explain the concepts of bookkeeping or sales tax or some other real-world mathematical task to a second grader, just so you can start talking about addition and subtraction? Ugh. One and one is two, little Johnny. Write it down.

Unfortunately, that elementary school experience is where most adults learn how to teach. We get a little locked into that approach early on, and because most people don't *think* about how they learn, they don't question the approach. Just show someone how to do something, and they'll learn it. Sadly, adult brains are less absorbent than kids'. Adult brains, as I've written in previous chapters, don't *want* to learn as much. The brain is busier with day-to-day nonsense, and before it learns something it sits down and asks itself, "seriously, do I *need* this to survive, or can I ignore it?" Humans don't have conscious control over that behavior - try to show something to our brains that we don't need to eat and breathe, and it just slides off. The more abstract the concept, the less likely our brains will seriously care. You can point your eyes and ears at something for as long as you want, but your brain just won't soak it up unless there's a reason.

Now, the way most training materials try to combat this is by *giving* you a reason. "Frying pans get hot," a book might tell you, "and so you have to use a potholder when you pick one up off of the stove." Problem and solution in one neat sentence. Except that *telling* your brain to care is not the same as *making it actually care*.

Remember, this triage behavior of your brain - wherein it only pays attention to things that *it* thinks are important - isn't conscious. You can't bypass the brain by reasoning with it. So every time a course does the typical problem-solution statement approach, the brain often doesn't pay attention, and the material doesn't stick.

You know what makes something stick? Experience. Put your hand onto a hot frying pan and your brain will assuredly remember the pot holder trick the next time. You can't *present* the problem to the brain. The brain must *experience* the problem - but once it does, it will *seek out* the solution, and it will by God *remember* that solution.

Experiences don't need to result in physical pain, though - any "failure" situation will do. Simply frustration at not being able to accomplish a task. Deep curiosity about how to do something better. Anything, really, that makes the brain *want* a solution will work.

This is why narrative sequencing is so vitally important in adult learning materials. In the previous chapter, I wrote that I use sequencing to ensure I'm not foreshadowing material; I don't introduce something until I'm ready to talk about it more or less in full. That's still an important rule, but a companion rule is that your sequencing also needs to allow for natural, self-discovered segues from unit to unit (whether those units are book chapters, class modules, or something else).

For example, suppose you have a book chapter that discusses cooking basics. You explain how to select an appropriate pot, how to fill it with water, and perhaps how much salt to add to the water (we're cooking pasta, here). You wrap up the chapter by explaining that the pasta will only cook if the water is hot... creating the obvious question of how to make the water hot. The next chapter covers boiling the water, including safety precautions, how to watch for when the water is boiling, and so forth. Of course, once it's boiling, you need to dunk the pasta in it... which is the problem covered in the next chapter. That's obviously a lightweight example, but the idea is to sequence learning material so that each unit ends in an obvious problem statement - one the student's brain can immediately agree with, and one that the student's brain may in fact have already been asking.

Some important take-aways, here:

- You aren't giving the learner a problem statement - you're asking them a question, which lets them form their own internal problem statement. "But how can we get the water hot?" gets their brain in agreement. Simply stating the problem doesn't accomplish the same mental agreement.

The Nine Principles of Immediately Effective Instruction

- You have to sequence material really, really carefully, so that you can flow from one instructional objective to the next by means of a series of learner questions. You do have to "set up" the learner for those questions, but it's important for your material to make the questions natural and obvious. Yes, you're leading the witness - that's the point.

This can be *difficult*. It requires a lot of communication skill, writing ability, and in live classes a lot of presentation ability. Instructors are mentally geared to answer questions, not to ask them - and certainly not to leave questions hanging, which is exactly what you have to do.

Let me offer a more realistic example. I often teach about a product called Windows PowerShell. One of the things you can do with PowerShell is construct computer commands. When covering that topic, I'll start students with a fairly basic command that uses some hardcoded piece of information, like a computer name. Students invariably ask, "okay, but how can I make that computer name easy to change, if someone wants to run the command on a different computer?" Well, that's the next chapter. Then it's, "how can I *require* them to enter a computer name when they run the command?" Next chapter. "How can I document the fact that they have to enter a computer name, and provide examples of proper command use?" Next chapter.

You'll never be able to set up a sequence like this until you've taught the material a few times. Teaching lets you try a sequence, and lets you pay attention to students' natural mental progression. By observing that, you can re-sequence the material so that you're getting the questions in students' heads, and presenting material in a natural order. That's why some of the best instructional books are written by people who've taught that same material to a live class. A teacher who's got a good delivery sequence for the classroom is a teacher that's

had a *bad* sequence a few times, and who has worked through and re-sequenced the material.

I used to own ferrets. The thing about ferrets is that they're latrine animals, which means they tend to do their business in the same spot. Unlike cats, they won't seek out a place to make their toilet - they just kinda pick a place they feel comfortable with. That means *you* don't get to decide where the litter pan goes. Instead, you must observe where the ferret backs his little butt into a corner, and plop a litter pan in that spot. You're observing the creature's natural tendencies, and leveraging that to get what you want. You aren't going to re-train the ferret, I promise. But you can still get it pooping in a pan, if you play along with its brain.

That's pretty much how you need to sequence your instructional material. Come up with a sequence that you think works, then try it - ideally with a live class - and observe where the students' brains seem to wander. Don't try and wrest those brains onto a course of your own choosing. Sequence your material so that you're satisfying the brain's own wandering. Obviously, different students will "wander" in slightly different directions, so you'll need to sort of look for large trends and go with those. You'll also need to set up your students' brains.

For example, when I talk about building PowerShell commands, I start - as I wrote previously - with a simple command that has a static computer name. In my materials, I emphasize that. "This command only targets the computer named FRED. If you decide you want to target a different computer, you have to open up the text file containing the command, and you have to go to the computer name, and you have to type in a new computer name. Then you have to save the file, and go back into PowerShell, and then you can run the new file." By this point, every student brain in the room is thinking, "what a pain in the butt that is." I've made them *feel the pain*.

They can picture themselves doing what I described, and they don't like it. "Isn't there any way to have it just *prompt* me for the name?" someone will quickly ask. Next chapter. Without my belaboring the point, some students may have wanted to know how to add a password to the command, and others might have been wondering what was for lunch that day. By setting them up, I got them wandering in a pretty natural direction - *some* of them were already wondering how to make the computer name easily changeable - and I got more of them wandering in that same direction.

I don't want to understate how hard this is. Most authors can fully understand the desire to sequence material in a logical fashion for presentation. *I'm going to show you this, because you need it to know this other thing, which you need to know before that last thing.* I call that "dependency sequencing," and it's a pretty obvious design consideration. But after you come up with that sequence, you need to make sure that it's also presenting the material in a *natural* order. Before you teach B, you need to teach A - but when teaching A, are students led to an obvious, natural question that leads them to be curious about B? Do they want - nay, do they *need* - B by the time you get there? If not, you're either going to have to re-sequence your material, or you're going to have to re-write your discussion on A to make the path to B much more obvious, natural, and desirable to students' brains.

If you don't make them *want* the next topic, their brains won't care. If their brains don't care, they won't learn it effectively. Period.

IMMEDIATELY EFFECTIVE™

Help students embrace new information by first helping them *experience* the problem that the new information solves. Explaining the problem in advance isn't the same as experiencing it.

You Don't Know Their Life

Just because you can tell stories doesn't mean you can tell *my* story. So don't try.

This chapter is going to earn me some evil looks, I know. It's because I'm about to take a time-honored instructional design trick and tell you not to do it. It's a trick developed and espoused mostly by people with one or more PhDs hanging on their wall. They're probably smarter than me. But they're wrong on this one. I've actually tested it, and I'm right-right-right.

Instructional designers and authors love to write scenarios. They'll even go so far as to call them *real-world* scenarios. In many instances, these scenarios are a required element when you're writing a course for someone who has a pattern or model. For example, when I've written courses for Microsoft, every hands-on lab absolutely *must*, full-stop-and-no-debate, be introduced with a "real-world scenario."

You are an administrator for Consoto ["Contoso" is one of the most common fake company names in Microsoft's courses]. You and your colleagues frequently create new user accounts in the company's Active Directory domain. This process takes several minutes per user. The new user information is provided to you by the Personnel department in the form of a CSV file. You want to automate the new user creation process by using Windows PowerShell.

First of all, nearly anyone who's been through a couple of these courses sees one of these scenarios and just skips it. It's like getting a new Blu-Ray player. You flip open the manual, there's a page of electrical safety

warnings blah blah blah, nevermind-I'll-just-start-plugging-things-in.

The *intent* of these scenarios is laudable enough. The idea is to give the student something that they can relate to. "Here's *why* you want to learn this." That falls neatly in line with what I wrote in the previous chapter, right? *Wrong*. These scenarios miss the bit where you *can't tell someone's brain why it cares*. The student's brain needs to *construct* reasons for caring - it can't be told. "You're an administrator for Contoso." No I'm not. "You frequently create new user accounts." No, I don't. We have another team for that. **YOU DON'T KNOW MY LIFE!**

And that's really what it boils down to. As an author, you can't lump all of your students into a single pile. In order for a scenario to work - in order for it to create context for a conversation, and in order for it to give the learner something to relate to - the scenario must be something the student recognizes. There is no way in the world you can write something real-world that *every* learner will relate to.

When I first started teaching a particular automation technology, I needed to show some real-world examples in my demonstrations. I was teaching people who worked in a Microsoft-based technology environment, so I had a lot of things to choose from. I could have chosen messaging, for example, because every environment has e-mail, right? But messaging systems typically have dedicated administrators, in all but the smallest environments, and I knew I'd have students who weren't even allowed to touch the messaging system except as an e-mail user. So I figured I'd use Microsoft's authentication service, Active Directory, for my examples. Every Microsoft-centric environment uses Active Directory, and it touches nearly every Microsoft-branded business technology component. Whether you need to send an e-mail, open a shared document, or print

a file, you're using Active Directory. Every student in my class should know what a user account was, and how one was made, and so creating new user accounts would be a good example. Boy, was I wrong. Literally two-thirds of the students in my class would immediately go glassy-eyed the minute I started talking about creating user accounts. "What's the matter?" I'd ask. "I don't really do that in my company," they'd say. "Well sure, but you know it has to be done, right? You know this exists?" They'd shrug. It was abstract for them. What I was showing them had no bearing on their personal, daily lives, so they didn't care. You can't *make* the brain care. It cares about stuff that affects it. It ignores the rest.

And that's why these fakey scenarios fail. If you're going to show me how to change the toner in a copy machine, I'll care if that's something I do as part of my job. If I don't do that as part of my job, no amount of storytelling is going to make me care. I'm just going to sit and wait until you get to something I care about. And if I *do* change toner as part of my job, I sure as heck don't need some little story to get my attention. I know what the problems are already from personal experience. So the scenario is either useless or redundant.

The next argument is usually, "well, it might not be useful, and it might be redundant, *but it doesn't hurt to have it in there.*"

Disagree.

When I was a kid, we'd sometimes go out to a restaurant for dinner. I can't count the number of times Mom would say something like, "don't fill up on bread and water before they bring out our food." Mother, bread *is* food. Tasty food. And you're telling me *not* to drink water? What kind of parent are you, anyway? Now, of course, I realize that what she was saying was more like, "bread has practically no nutritive value, and water has zero, but they take up room. Your little tummy only has so much room, and if you fill it up with those, you'll be hungry

again by the time we get home, and I am not making you a sandwich."

I am a firm believer that the human brain can only take so much input in a given period of time, and there's quite a bit of casual experience and formal research to back me up on that. What I refuse to do is give my students bread and water while I've got such large entrees already set to come out. Sure, *most* of my students will skip the scenarios and get right to the main course - but *some* will read the stupid things. The scenarios will not enrich the learning experience, but for some students the scenarios *will* take up brain space. Precious space I'm going to *need* for actual learning. One reason I'm a successful instructor - and I gauge success by student retention and application, not by whether or not students like me - is because I'm *concise*. My business partner and I named our company "Concentrated Tech" because we know technical people value concise information. Even a single wasted word, one that adds no value to the conversation, is painful to me. Thus, I hate fakey scenarios. They add nothing. They are *waste*.

What's funny, for me, is that you never see fakey scenarios anywhere else. We don't use them in K-12 education. They're rare in undergraduate courses. The assumption in both cases, I suppose, is that you're there to learn, and you'll just learn without a little story that's supposed to make you feel "connected" to the material. You don't see scenarios in any other kind of communication aimed at adults, either. Commercials - which focus on delivering a concise message in an extremely short period of time - spend time extolling product benefits and the like. There may be a *story* behind the commercial, but you don't have to get a preparatory paragraph first. Newspaper and magazine articles - which also aim to be concise - don't start with a scenario. So where did we get the idea that scenarios are necessary?

Scenarios aren't bad in all forms of learning. Some people get a little carried away. For example, I saw a quote at an instructional design seminar: "Creating thought-provoking scenarios stretch the learner's imagination and help them journey through varied emotions until they reach their learning peaks." Zoinks. That's a lot to lay on the shoulders of a little story. But the general idea behind scenarios is to help a learner imagine themselves in a particular situation. If they can do so, the reasoning goes, then they'll "feel the problem," and be more receptive for the solution. It's basically the approach I outlined in the previous chapter, except that I think it's a false presumption. For one, a lot of adult learners don't *have* a great deal of imagination. And, particularly in vocational-style training for a particular profession, learners aren't showing up to learn about what *other* people might need, they want to find solutions for *their* day to day problems. If you have to give me a narrative that helps me imagine myself in someone else's shoes, then go find that someone else and teach *them* your class. I came to class looking for something else, and if you're going to teach it, I won't need to *imagine* myself in that situation. I'm *in* that situation, for real, every day.

This is why I tend to advocate for an approach that puts the learner in a hands-on situation that lets them *experience* the pain (or whatever), instead of telling them a story where they *imagine* themselves in a situation. For example, let's say that you're not especially interested in being a garbage collector, but for some reason I need to teach you about more efficient trashcan design. I could start with a scenario: "You're a trash collector. Your manager tells you that you need to cover more houses in a smaller amount of time, but existing garbage can design requires you to spend almost 90 seconds at each house. You would like to design a more efficient trashcan."

Got that in your head? Can you smell the trash, and feel your muscles burning as you lift those unwieldy cans? No?

Another approach would be to walk you outside and have you empty a few inefficiently designed garbage cans. "Here are ten cans. You've got five minutes to empty them. Go." Afterwards, we could discuss - as a class - what made the task harder, and how a different can design might make it easier.

Which approach do you think would be more effective?

I'll conclude with a final appeal for practical engagement over scenarios. I'm sure you work for, or have worked for, or at least know someone who works for, an organization where managers have made some pretty stupid decisions. Blindingly, obviously bad decisions. And I'm sure that, upon observing those decisions, you've thought to yourself, "wow, if that person had ever worked in this job, they'd know how stupid that is." Exactly. That's what happens when you ask someone to *imagine* themselves in a situation, versus *putting* them in that situation. You might tell a story to a manager about your job, but it'll never be as concrete or as valuable as if they just *do* your job for an hour.

So as you design your instructional materials, try to avoid an approach that relies heavily on scenarios and imagination. You're adding to the student's reading load, and you're not adding a lot of value to the class. Scenarios don't teach much, and they don't really get the learner's brain to "buy off" on the upcoming instruction. They're a distraction, and an ineffective one in most instances.

IMMEDIATELY EFFECTIVE™

Artificial scenarios won't resonate with *every* student, and they create a distraction. Create an overarching narrative that involves students *experiencing* problems, and they'll embrace the solution without needing a scenario.

Shut Up

Concise.

Instructors talk too much.

And, along the same lines, authors write too much.

Really, I should just end this chapter there. But I'm being paid by the word (not really), so let's expand on that. When I've taught a class a few times, or written on a subject a few times, I find myself becoming more and more concise. I eliminate stray material, because frankly the extraneous stuff isn't what my students came for. My most recent book was under 600 pages and my publisher was annoyed, because they'd priced the book for 800 pages and had to adjust. But concise is better. Write less. Talk less. Force yourself to be concise.

I think this comes down to objectives. If you haven't decided up front exactly - in a good amount of detail - what you're going to teach, then you're going to teach too much. Perfect example: a colleague was using one of my *In a Month of Lunches* books to deliver an in-house class. After his first day, he tweeted, "don't get outside the chapter! you will not finish!"

This happens to a lot of instructors and writers, and it happens all the time. Instructors - good ones, at least - are enthusiastic about their topics. They *want* to share. They want to share as *much as possible*. They cram in little extra tips, tidbits, and trivia. They become a veritable firehose of information, blasting into an unsuspecting crowd.

It's too much.

Humans simply can't digest that much - something I touched on in this book's first chapter. As I wrote there, you need to decide in advance exactly what you *need* to

teach. What skills are you going to enable? What is the absolute minimum amount of information you *must* impart in order to achieve those goals? Can you reasonably and *effectively* do so in the time available? You're an expert in your topic, but you amassed that expertise over a period of, what, months? Years? You can't impart it all in a single class, or a single book. *It's too much*. So, once you've narrowed down your learning content into the absolute minimum, focus on also minimizing the number of words it takes to impart that knowledge.

An example: I've had to teach people how indexes work in Microsoft SQL Server. I've worked with that product for over a decade, and I know how indexes work, but I hadn't really *taught* the subject at the time. So I watched a few other instructors, most of whom followed the material in Microsoft's official curriculum for SQL Server. The average index explanation took over an hour. Ugh. I didn't have an hour, and watching those other guys, quite frankly, made me bored (I don't have the attention span to sit in a class myself). So I started looking at the material more carefully, narrowing down exactly what you *needed* to know, and trying to cut down on the sheer verbiage. A picture, I've been told, is worth a few hundred words at least, and so I created some diagrams to help explain some of the stickier bits. The first time I taught the class, I took about 45 minutes to cover that subject. On subsequent deliveries I pared it down to 25-30 minutes, which is kind of a magic delivery time number for me. I took the time to question my students carefully, and to re-question them on subsequent days, to see how much they'd retained. They retained a lot - and most of them could explain the topic back to me, or to the other students, on request. As an experiment in a later class (never take a class with me unless you're okay being an unwitting guinea pig), I reverted to a longer, almost 60-minute delivery - and retention fell way off.

In a previous chapter I mentioned a magazine editor I worked with, Keith Ward. He's the one who taught me that learning to write an 800-word column, instead of a 2,000-word article, would make me a better writer. He was right. *Concise* is the watchword of training.

Let's use an analogy: imagine that you like chocolate truffles. Imagine, too, that you can choose between one of two boxes. The first box contains ten truffles. The second box contains ten truffles and 5,000 ping pong balls, wrapped in foil just like the truffles. Which box would you prefer? Probably the one that's more concise - that contains *just* what you want, with no extras. Right?

Which is why this is the shortest chapter in this book.

IMMEDIATELY EFFECTIVE™

Ask yourself what you can safely eliminate from your delivery while still meeting your learning objectives. Deliver as little material as possible to accomplish your goal, and *don't overreach your goal*. Just because you have extra time or pages doesn't mean you *must* use it all.

Be Profound

Boring instructors *go to hell* when they die. Possibly *before* they die.

I'm probably most well-known for teaching a Microsoft product called Windows PowerShell. PowerShell is an odd product. It's intended for administrators, but it uses a very different approach to administration than many of them grew up with, and there's been a lot of resistance, by some folks. The problem is, Microsoft is really moving toward that approach, and folks are putting their career in serious jeopardy by not getting on board. It's like making steam-powered cars after the invention of the gas engine - you're just in denial. So I came up with a little catchphrase, which has become a mildly famous proverb in the Microsoft world: "Learn PowerShell, or learn to say, 'would you like fries with that?'" It's a deliberately demeaning, dramatic statement intended to elicit a chuckle, but also to make a very important point. It's easy to remember - like a good song lyric, it bears repeating and becomes a meme unto itself.

There's a component in PowerShell that produces textual output, which actually flies in the face of what the shell is and does. That component has some specific uses, but people tend to stumble across it and use it inappropriately. This "Write-Host" command, as its called, actually creates a hurdle for learning the product, because it gets you moving in a direction that makes just about everything harder than it needs to be. So I'm also mildly famous for the saying, "every time someone uses Write-Host, God kills a puppy." There are t-shirts, even. The statement isn't even entirely true, because as I've noted there are legitimate uses for the command. But it's a humorous, dramatic - *profound* - statement that sticks with you. After hearing it, folks think twice before they

The Nine Principles of Immediately Effective Instruction

set down that direction, which was the goal. Later in the class (or book), I'll expand the discussion and talk about when the command is good and when it's inappropriate, but the profound statement sets the stage and creates a lasting impression.

The point is that over-the-top, profound statements are an excellent learning tool, provided you use them sparingly, and only in places where a memory would otherwise be difficult to create. The more abstract the concept, the more likely a dramatic, possibly humorous statement will be effective.

Another component of PowerShell is its built-in help - essentially, its operating manual. The help file has its own complex syntax involving way too much punctuation. For example:

Do-Something [-computername <string[>]]

It's a bit abstract, and that makes it hard to remember. The item in <angle> brackets tells me that the computer name must be a string of characters, and the [] indicates that I can provide multiple names if I want. The outer [brackets] indicate that the whole thing is option - I don't have to provide a computer name.

Explaining this in a live class is especially hard, because - unlike in a book - students can't "rewind" and review something you've just said. You end up trying to rely on repetition alone to drive the point home, and that isn't always effective with everyone. An instructor I know has used profundity to make everything easier to remember. His approach to "profound" isn't a catchphrase, though - it's almost pantomime.

"What's the parameter name?" he'll ask. "Computer name!" the class responds. "HOW DO YOU KNOW?" he'll shout. "THE DASH!" they all say, and every student holds up one hand in a horizontal position, miming the hyphen that precedes parameter name. "What kind of data does it take?" he'll ask next.

"Strings!" they answer. "HOW DO YOU KNOW?" And the students, to a person, hold up their hands in "angle bracket" positions, and cry out, "cha-hooah-hooahs!" which is the name the instructor gave to angle brackets. "Can it take more than one?" he asks last. "YES!" they yell, holding up their hands in [] square brackets. It's a delivery technique I can't quite bring myself to steal, but damn is it effective. He also leverages repetition, because every new command he shows them involves a repeat performance, complete with pantomime. The hand motions, the repetition, the performance, and the nonsense "chahooahhooah" name all combine to make for a memorable delivery of an abstract, difficult-to-remember concept. He does all this on the first day of class, and what's impressive is that one days 2, 3, and beyond, students are still playing along, and *getting it*. When they get stuck in a lab, all he has to do is hold up the "cha-hooah-hooah" hand sign, and the student says, "ah, right!" and they're off and running. Profound. Effective.

If you think about it, most of the mnemonics we're given as kids, to help with rote memorization, are also profundities. "Superman Hates Eating Onions" is ridiculous, but memorable when it comes to the names of Great Lakes (Superior, Michigan, Heron, Erie, Ontario, I think). "Man Very Early Made Jars Sitting Up Neatly, Period" helped me remember the names of the planets, nonsensical as it is.

You do have to time your profundities carefully. I find that I get about two per day, max, before they stop being affected - and I use that limit as I'm sequencing material to make sure the over-the-top statements are appropriately spaced. If there's an easier, or less dramatic, way to help students remember something, I'll explain that. And, because profundities aren't always 100% accurate, I do have to take the time to explain the underlying meaning and any exceptions. I'll never give that explanation in close proximity to the profundity,

though, because doing so diminishes the impact, which was the whole point of the thing.

"Profound" also doesn't universally mean "silly." Sure, there are some silly things - "ca-hooah-hooah" - that can be used in an impactful way, but they work in large part due to contrast. In other words, they work because the rest of your delivery is more serious and professional, so that one silly thing sticks out... profoundly.

Profound also works in books, although obviously you lose the opportunity to use pantomime and other physical delivery tricks. You become limited to phrases, and it is a lot harder to make a phrase stand out from all the other phrases around it. I tend to use *italics* a lot for emphasis, as you've seen, and **boldface** is obviously an option. However, in many kinds of writing those elements are reserved for other things - boldface is often used, in technical books, to refer to user interface elements ("click the **OK** button"). So you have to get a bit more creative to make your profound statements really stand out. I've found that, rather than using visual styling, I get more impact by building those profound statements into a short story. Just a paragraph, maybe three or four sentences; I'm still acutely aware of wasting space. But because I *don't* tell many stories in my writing, when I *do* tell one, it stands out for just that reason.

So why does "profound" actually work? As with most of what I've written in this book, it works because of the way our brains work. Humans tend to build their worldview around markers and milestones. We celebrate annual events like birthdays, we navigate largely based on landmarks ("turn right at the post office"), we often shortcut rote memorization by using mnemonics. Making profound statements not only provide contrast with the rest of an instructional delivery, but they tap into our brains' desire for landmarks. When they're big,

memorable statements, our brains grab on and tend to not let go easily.

IMMEDIATELY EFFECTIVE™

Make an impact - but do so judiciously. People - aside from cable news anchors - can't take constant shock-and-awe, but a well-timed (and humorous) jab can make something really stick.

And Now for the Top Ten

<p>You only need to remember three things from this chapter.</p>
--

You only have to look at modern media to see that people are natural collectors. Hoarders, even, although most of us try to control our acquisitive urges. Video games offer badges and achievements, scavenger hunts remain a popular pastime for kids and adults, collectors of all kinds invest considerable time and money completing their collections.

We can leverage that behavior in instructional design. Now, most instructional designers will agree because they've always been attracted to numbered lists. They'll tell you that people work well with numbered lists, especially when those lists describe some sequence of tasks that the learner must perform. That's all true, but it isn't what I'm talking about. A numbered list doesn't encourage *collecting*. It's like giving someone a complete set of postage stamps - there's no actual collection activity. Collecting requires the collector to expend effort to obtain each item in the collection, and there's an implied period of time between each acquisition. During that interval, the brain savors its most recent acquisition, and greedily anticipates the next one. What you have to do is set that up properly, by telling the brain - that is, the learner - that a collection opportunity is coming up. You clearly communicate how many items will be collected, and you highlight each collection opportunity. Learners won't consciously perceive all of this, in most cases, but their brains definitely will.

For example, one topic that I've written on has a number of features designed to help people learn to use the product. Specifically, there are three of those features. I

typically have an instructional unit dedicated to each of those features, and so I start with an overarching narrative like, "there are three discoverability commands that you need to know - and if these three are the only commands you remember from this entire course, then you'll be fine." Then, as I introduce each element, I call attention to it: "this is the second of the three commands I told you about. Remember that the first was named 'Help,' and this one is called..." and so on. That allows me to use repetition - I get to repeat the name of each command as I introduce new ones - as well as engage the collector portion of learners' brains.

That approach also does something else for me. While brains will be (subconsciously) eager to collect when there's a known quantity of items to acquire, learners' brains can get a bit (consciously) anxious when there's an *unknown* number of items. That's especially true when you're teaching an unfamiliar topic. A lot of adult learner anxiety comes from a simple fear of the unknown: "I don't know how much there is to learn, I don't know what it's going to involve, and I don't know if I can do it." It's like getting onto a thrill ride in a theme park, without being able to see it first. In that case, the anxiety is part of the attraction; in class you want to avoid it. So my approach tells students that there are (for example) three important things coming up - and if they can learn those three, everything else is just icing. The brain immediately decides, "ah, three things I can definitely handle, and it's safe to ignore the rest." Now obviously, that's not *entirely* true - the rest of the class is hardly "optional." But because I sequence the material so that these three critical pieces get covered up front, I can continually reinforce their use throughout the class. Additional learning items almost take the form of practical exercises in using the three critical things, so while they're not optional, I'm not entirely untruthful when I say that those three things *are* the main bits the learner needs to remember.

Some authors have argued that stating a list of learning objectives will trigger the "scavenger hunt" reflex, but I don't entirely agree. In any course you write, you're going to have a variety of objectives, none of which are of the same "flavor." In other words, a class on car maintenance might involve changing oil, changing wiper blades, adding gasoline, and so on. But those are all very different activities. You can't necessarily point to them and call them a *collection*. What you could do, perhaps, is say, "you're going to learn to change three kinds of fluid," and then get the student to "collect" those similar skills. However... what's the point? The student *knows* they're going to learn those things, because presumably they were listed in a course abstract or outline, right? You don't have to trick the student into remembering the objectives, because for the most part you don't care if they actually *remember* the objectives themselves. You don't need the student to recite a list of things they're going to learn. So I don't rely on objective to create "collections" for my students.

Enabling skills are what you want students to collect. "Now, to be really successful as a car mechanic, you're going to need to know how to use a variety of tools. But there are really only four types of tool that you need to worry about - if you can get those four, the rest will be easy." You wouldn't list "use a crescent wrench" as a course objective, because that isn't a useful job skill. The job skill might be "change the oil," with the crescent wrench being a means to that end. But by using "tools" as a collection, you can get students' brains to start looking for those things. You reassure them that they *can* handle what you're about to throw at them ("...only four types of tool..."), and you give yourself an opportunity for important repetition ("the third tool is a socket wrench - it performs the same basic task as the crescent wrench and adjustable wrench you've already learned about, but...").

Don't overuse collections. Like any leveraged behavior, it works best when engaged as a strategic element in a class. Most people, for example, only collect one or two kinds of things. Maybe it's comic books, maybe it's stamps, maybe it's petrified bugs - but it's rare for a person to engage in "collector" activity in a number of different things. That's partly because a collection represents a kind of "to-do" list to the brain, and it'll get anxious if it has too many "to-dos" stacked up. By saving the "collector behavior" for the most crucial information in your course, you'll help the brain focus. It does mean that you, as an instructional designer and author, need to really triage your information and decide what's *truly* crucial. I always ask myself, "if I could only teach someone three to five things in this topic, what things would give me the biggest bang for my buck?" I might, for example, decide to *only* teach the student about crescent wrenches, and to downgrade my focus on similar tools like adjustable wrenches or sockets. Those are nice, to be sure, but I can get *most* of the job done with a basic crescent wrench, and downgrading the time spent on other wrenches lets me involve a couple of other critical components, like screwdrivers and hammers.

Remember that *you can't teach everything*. Every course will always omit something that would be *great* for the student to know - and most courses will have to, by sheer practical necessity, omit *several* great things. I spend more time doing triage on my material to make sure I can do a fantastic job teaching what *will* fit into the course's allotted time. Focusing on "collections" of information helps me do that. "Man, there are ten things I'd really like to show them - but I'm going to cut that list down to five. What five things are the most awesome, the most useful, and the most important?"

IMMEDIATELY EFFECTIVE™

Get your students' brains ready to collect information by presenting them with small collections to acquire. Try to sequence your material to introduce items in the collection over a short period of time, maximizing the brain's anticipation and satisfaction cycles.

You've Got 30 Minutes to Make This Real

Nothing cements knowledge like *using it* - but you have to get there quickly.

Human beings are doers. We're noteworthy on our planet for being one of very few species to create and use tools, but the fact that we're "tool users" is really a symptom of the fact that we *do* things. Our brains learn best by doing. Engaging our brains *and* bodies in something makes it real for us, and really makes new skills concrete.

Practical experience also serves another crucial learning goal: the opportunity to fail. After all, no matter how well-written a course, and no matter how great an instructor's explanations, and no matter how wonderful the demonstrations, when you do it yourself for the first time, you're likely to make a mistake. Our brains *really* learn well when we make a mistake, find out what we did wrong, and then correct ourselves on a subsequent attempt. Whether it's baking a cake, assembling a machine, programming a computer, or even drafting a last will and testament, it's the opportunity to try something, *fail*, and then succeed that makes for effective learning.

The problem I find with most courses and books is that they don't get students into hands-on experience quickly enough. Our brains are pretty terribly at queuing up information for later use; until we've *used* something, our brains are thinking, "this isn't useful, so I don't want to remember it." *Using* new knowledge gives your brain a context for it, and a reason to hang on to it. Too many authors and instructors forget that fact.

For example, suppose you bought a cookbook, and suppose that it started with a lengthy discourse on the

origin of wheat as a food item, going into deep detail on how grains are harvested, milled, and packaged. Yawn. Most readers would skip all that and get to the recipes, right? In fact, why even include that information? Well, the information *is* useful, and it *is* relevant. Knowing how wheat is milled gives you a greater understanding of how different flours are used in different recipes, for example. But your brain doesn't want to queue up all that background information for later use - it wants to start *doing* stuff.

Once you accept the maxim that students should get no more than about 30 minutes of education before they have a practical experience, you force yourself to sequence material a bit differently. Background material that just isn't relevant gets dropped. Conceptual information that *is* relevant gets introduced as concisely as possible, to save time, and tends to be taught "just in time," right when the student needs to know it. In my cookbook example, you might dive right into a recipe for cake, and learn that a specific kind of flour is preferred for light, airy cakes, and learn a bit about the milling that makes the flour right for that task. The next recipe might be for bread, and might briefly discuss why that requires a flour that's been milled a bit differently. The discussion on how grain is actually harvested gets dropped, because while it's *interesting*, it doesn't really lead to better baking.

The idea of dropping information is a controversial one, and it's probably the one principle in this book that I argue about the most when I speak to instructors, authors, and instructional designers. Instructors want to be a fire hose of information. They want to cram data and facts into the course - and into students' heads. They believe students value "hardcore" learning where only the fittest survive. They speak of "crash courses" and other filled-to-overflowing approaches. The problem is, students don't retain much of that information. Students tend to retain only what they've been able to practice and

experience, on their own, within about 30 minutes of learning it. Jamming more information into a course does *not* make it more valuable. You *cannot teach it all* in one sitting. Instead, you have to - and I've probably used this word in every chapter up to this one, so I might as well use it again - *triage* your information. You have to pick the most important, impactful, and reusable information and teach *that*. You have to get students using it *quickly*, or you might as well have skipped it - because without rapid progression to practical experience, the material will be lost anyway.

My courses tend to consist of short "sprints" of lecture, coupled with demonstrations as appropriate. A sprint might be 30 minutes, but it might just be 10. I teach *one* thing, and then get the students using it. When possible, I try to build on that one thing through subsequent sprints, and give students a chance to repeatedly practice each new skill through a series of hands-on experiences. Those hands-on periods might also be short; while I've certainly written labs designed for multiple hours, I've also designed lab experiences that take just 15 minutes. The course becomes a smooth flow of learn-try-learn-try-learn-try.

Of course, in a book you might think the whole "hands-on experience" thing is a little tough to achieve. After all, unlike an instructor-led class, books don't have labs and such, right? Wrong! In fact, a hallmark of my *In a Month of Lunches* series of books is that they *do* instruct students to "Try it Now" throughout each chapter, and most chapters conclude with a hands-on exercise. The chapters are written so that most adult readers can get through the text in 30-45 minutes, leading them directly to a self-paced "lab."

So how can you tighten course material to just a short "sprint?" Obviously, eliminating extraneous information is key. You can still provide "further reading" references so that students have a jumping-off point for future, self-

paced education and exploration, but tighten your course material to the bare minimum needed to accomplish your ultimate learning objectives. I literally sit and go through every single sentence in a course, once it's written, and ask myself how each one contributes to the stated objectives of the course.

Second, eliminate extraneous delivery. For example, I would rather a student do something on their own than demonstrate that same thing to them, so I'll frequently cut out demonstrations so that I can get students into a hands-on experience more quickly. I'll often limit my demonstrations to those that provide basic orientation. In a cooking class, for example, I'd quickly run through the tools the student was about to use, so that they could recognize and use them. I might demonstrate a whisking technique, but I wouldn't demonstrate an entire recipe. Yes, students will probably get something wrong on their first hands-on attempt - but that's the point. So long as you can provide guidance to help them self-correct, then they'll *learn*.

I avoid, as much as possible, trivia. For example, in a class on car maintenance, students need to be aware that engine oil gets dirty, and needs to be changed. They might need to know a few of the reasons why oil gets dirty, so that they can make good decisions about how often to change the oil - more frequently in dirtier environments, for example. They do *not* necessarily need to have an extensive education on the negative effects of leaving dirty oil in the engine. Yes, that's useful information, and it might well offer students more incentive to pay attention to oil changes. But you could probably provide that with a simple statement of fact: "dirty engine oil can cause major damage to the engine." Done. So really evaluate what you, as a course author or designer, *think* is important, versus what really *is* important. And, try to deliver the important material in the most concise fashion possible. When a statement or two can convey the information, there's no reason to

construct a multimedia presentation that takes four times as long to deliver. Less *is* more.

What if you've got some information to deliver that doesn't lend itself to any kind of practical experience? I'll boldly suggest that such a thing is not possible. If people can't *do* something with it, then they probably don't *need* to know it. Unless your course goal is to simply have people memorize facts (and what a boring course that would likely be), *everything* we know leads to hands-on experience. The nature of that experience might not always be obvious, so you may have to get creative and really think about it. For example, you might think that a course on family legal planning might not have a lot of practical experience opportunities - but isn't the whole point of such a course so that the learner can review, or perhaps write, legal documents? So that's the practical experience. Even when the "hands-on" material doesn't involve anything more than reading, it's a practical application of what was just taught - and so it helps cement things in the brain.

As I've written earlier, I always construct my courses by starting with the ultimate learning objectives, and I like to write those as things you would *do*. "After completing this course, you will be *able to do the following*" kind of phrasing. I then design hands-on experiences around those objectives, because hands-on - *doing things* - is how we learn best. The course material then simply sets students up for the hands-on experiences. Because the hands-on bits embody what I wanted to teach, I don't need to teach anything that isn't used in those practical experiences. The material practically writes itself from that point.

IMMEDIATELY EFFECTIVE™

Sequence material so that students are introduced to a very small number of items, and then immediately given a chance to put those to use. Avoid any lengthy instructional sequence that doesn't lead to hands-on experience.

Afterword

I'll emphasize something that I wrote in this book's Foreword: I'm not a trained instructional designer. I don't have a PhD - heck, I don't even have a college degree. What I do have are a couple decades' experience in communicating with adults who are trying to pick up new skills in their field. I also have a solid amount of experience with traditional instructional design, because I've watched it fail people over and over. I've watched it unnecessarily constrain authors, and I've watched students throw up their hands in frustration.

You can't, the saying goes, teach an old dog new tricks. The presumption behind that adage is that old dogs are set in their ways, and don't want to do anything new. When it comes to old - or at least adult - humans, I don't think the adage applies. Maybe *you* can't teach an adult human new skills, but *I* definitely can. All it takes is close observation of how their brains are working, and what their learning prejudices are, and a willingness to cater to those. I know I'm not going to change the way adults learn - but I can change the way I *teach* to better fit their mold.

Some of this book has been about pure instructional design, which is mainly about sequencing and triaging material. But instructional design also strays into instructional *delivery*, either by written or spoken word, and I've strayed over that line myself. *How* material is presented is just as important as what you present, and it what order you present it. In fact, I'll state outright that my nine principles only work in combination; leave out any one of them and you won't get the results you want.

I'll also leave you with a tenth principle: *Watch and learn*. Observe your students. Look for that knowing look in their eyes - or that glazed look an instructor fears so much. Pay attention to what you're doing. Teach with *deliberation*, and don't be afraid to rearrange material, change your delivery, or try another analogy with the

The Nine Principles of Immediately Effective Instruction

next class. Do so with purpose, and observe the results. Note what works, and what doesn't. Too many instructors just spew information into the room without really paying attention to the uptake, and without being willing to change their approach so that students "get it" more quickly and more effectively.

And above all, remember that every word out of your mouth, or every letter typed, must lead directly to students who can return to their jobs and be *immediately effective* with what you've taught. Otherwise, you've failed.

Go succeed.

Don Jones

July 2013

The Nine Principles of Immediately Effective Instruction

Adult education, particularly technology training, suffers from a lack of attention to instructional design. Instructors focus on technology, and on task sequences, rather than on ordering and presenting material in a way that an adult brain can best assimilate it. We teach grown-ups the same way we do ten-year-olds – and are surprised when those adults are less than successful at retaining and applying what they've allegedly learned. Master instructor Don Jones, author of more than 50 technology books, has taught tens of thousands of adults to be *Immediately Effective*[™] in a variety of technologies. Jones' approach focuses on careful sequencing of data and leveraging the natural learning tendencies of the adult human brain, enabling students to immediately apply new skills and techniques the minute they leave the classroom and return to their job. In this book, Jones shares his nine principles of instruction and instructional design, enabling any courseware author or class instructor to help their students learn faster, learn better, and learn more.