STORYCODE
COMPUTER PROGRAMMING THROUGH LITERATURE

SAMPLE LESSON CONTENT GUIDE:
"The Lady or the Tiger?" and Binary

Lesson 3 of 3 in module
Computer Programing Concept: binary
Text: "The Lady or the Tiger?" (adapted)

This lesson is the climactic learning experience in an early module of three lessons which introduce the concept of a *binary protocol, a way of doing something using only 2 elements*.

In the 1st lesson, the class shares a community reading of an adaptation of Frank R. Stockton's classic cliffhanger short story "The Lady or the Tiger?" with each learner writing an ending (or "version") of the story for homework.

In the 2nd lesson, the class explores mathematical methods for determining the number of versions of the story that are possible. (The 3 most accessible and powerful methods, "Brute Force", "Tree Diagram" and "Function" are reviewed in detail in this lesson.

In this 3rd lesson, the class connects the multiple versions of the story with binary, the elemental structure of computers.

Understanding this Sample Lesson Content Guide will be much more difficult without first reading the StoryCode adaptation of "The Lady or the Tiger?"

Content Guide Key:

Blue text represents topics and tasks identical to the notations in the corresponding lesson plan.

Bold purple text represents ideas and understandings that learners will hopefully contribute to the class.

Lesson Components:
(printable PDFs; see storycode.info)

1) StoryCode's adaptation of "The Lady or the Tiger?"
2) LorT Decision Point Chart
3) ASCII Chart (HW)
4) LorT CCSS and CSTA Standards

Title		Class		Term	Date
Class #	Aim: How is the coded plot of a story related to the way computers function?				
Do Now	(Groups) Use one of our 3 methods to show all the possible versions of "LorT?"				
Role		Collect HW		Return work	Collect Do Now
HW share	Anyone to read? Let teacher read? ***ID decision points of each version.***				

Share Do Now / Activity	Possible versions of "LorT?": ***It comes down to "decision points."*** 1) ***8 possibilities***: how do we know that? A] Brute Force (--> Code Box) ~ list on board i. ***ID decision points chronologically***; show horizontally: TK P L ii. Record decisions of truth or trust as T, falsehood or mistrust as F. iii. Record each 3 symbol code in its B2 position, lowest to highest. iv. ***Names for each version, along with death or marriage.*** v. How can you tell we're done? (***Can't. We need another method.***) B] Tree Diagram: ***1 (TK) dec. pt --> 2 (P) dec. pts --> 4(L) dec. pts --> 8 endings*** C] Function: permutations and exponential growth i. ***How does the # of T/F choices affect the # of possible versions?*** ii. Permutation Function ~ on board a. ***versions = (# of options per choice) raised to (# of choices) power*** b. ***f(x) = 2 raised to (x) power*** 2) What if the lady and the tiger could secretly switch places on their own? A] ***Show the versions for 4 decision points.***
Mini-Lesson 1	What do the decision points of "LorT?" have to do with computers? 1) Computers work with information in the form of 1s and 0s. A] 1s and 0s can mean almost anything. B] Meanings of 1s & 0s are determined by us. 2) Using 1s & 0s requires a "protocol" = method/order of doing something A] Define the protocol we used to code "LorT?" (our Code Box) i. Symbol: ***T = trust, F = falseness*** ii. Position: ***TK --> P --> L*** B] "binary" = system with only 2 parts (T or F in our case) C] "binary protocol"= ***method/order of doing something w/only 2 elements*** 3) Switch the Code Box to 1s and 0s. A] 1st identify the meanings of our symbols: T = True = 1, F = False = 0 B] Identify the meanings of our positions: 1st # = TK, 2nd = P, 3rd = L 4) Do we need to record the ending with another digit? A] ***No. We can always figure the ending out from the 1st 3 digits.*** B] Can you see any patterns for the endings? ***Single 0 --> death*** 5) TK-P-L Code Box = a binary protocol
Time? Mini-Lesson 2	ASCII, another Binary Protocol 1) What happens inside a computer when you press a key while typing? 2) The computer codes that letter as 1s & 0s then shows it on the screen. 3) Only the computer user sees this info. as anything other than 1s & 0s. 4) Hand out ASCII chart. A] = American Standard Code for Information Interchange B] = ***a binary protocol for the computer to understand the English alphabet*** C] 1 set of codes for lower case letters, 1 set for upper case letters 5) Does anyone know a simpler alphabetic binary protocol? ***Morse Code***
HW	Write your name in ASCII.

This page and all the contents of the StoryCode Curriculum copyright© Lev Fruchter 2013

Aim: How is the coded plot of a story related to the way in which computers function?

Remember, the class defined *plot* when we read "The Lady or the Tiger?" together.

Do Now (Groups): Use 1 of our 3 methods to show all the possible versions of "LorT"

> [A "Do Now" is a task that's on the board when learners arrive in class. This is a useful tool for making sure class begins with purpose and on time. A Do Now can introduce a new and interesting idea or review something learned the previous day or for HW. Learners' heads can be anywhere when they arrive, so a Do Now should focus the attention rather than wrack the brain. If you can arrange your room in learner tables of 3 to 5, they can engage the Do Now each day in prearranged small groups, exercising teamwork skills and, most importantly, stimulating conversation about the topic.]

Each learner should have the 3 methods explored in the last session in their notes: Brute Force, Tree Diagram and Function. The groups in which the learners start class should compare notes, decide which method is best to use for the Do Now and then cooperate on producing an exemplary demonstration of the number of possibilities in the story.

During this time, collect any new HWs and, most importantly, return the endings to "The Lady or the Tiger?" which the learners wrote for HW after the 1st lesson and handed in during the previous session.

Share HW: Anyone to read their version of "The Lady or the Tiger?" Let teacher read?

> [All the class benefits in so many ways from sharing creative writing aloud. Do whatever you can to establish an environment in which learners are unafraid to present their work to the group. Build a safe space of mutual respect and support early and it will pay dividends throughout the entire year in every aspect of class. *Reaction is wonderful; ridicule is unacceptable.* Be ready to interpose your strongest teacherly presence in the face of any derision or mockery.]

If no learners volunteer to read their own work, offer to read anyone's yourself. You could even ask how many people would like theirs read, collect them, shuffle them and then read them so that the class cannot be sure who wrote which piece. *Don't be overly literal when reading the learners' endings aloud!* It's helpful to massage clunky bits of grammar into a smoother recitation, so long as you don't neutralize the writer's voice. Remember, this is the first draft of a creative piece in which the content, not the form, is most important.

If no one offers to share their composition in any way, read aloud 1 or 2 that you have prepared ahead of time or make a couple up on the spot or adapt the examples given in Section 1.A.iv. below.

ID decision points of each "version".

You may want to preface the recitations by telling learners to raise their hands or even remark out loud whenever a critical part of the plot is decided. Today's biggest understanding is that the decisions made by each of the 3 characters in "The Lady or the Tiger?" determine not only the ultimate fate of the lover, but also the particular *version* of the story that is taking place. Even if learners interrupt the recitation when a character makes a decision, they are helping everyone to move toward that understanding.

After each composition is read, ask the class if the author correctly determined the lover's fate in their version of the story. Allow time for debate about each case.

Share Do Now/Activity: Possible versions of "LorT": ***It comes down to "decision points."***

Now that you've drawn attention to the characters' decisions in a few versions, it's time to puzzle through the mathematical structure of the story. Learners may be eagerly calling out the number of different versions of the story they believe are possible, and it's helpful to write down each suggested number in a corner of the board. Much more important than any number, of course, is the computation that leads to it. Each of yesterday's 3 methods should be shared to illustrate ***the 8 possible versions*** of "The Lady or the Tiger?"

A] Brute Force (--> Code Box) ~ list on board

Also known as "proof by exhaustion", this method is simply to list every possible case that fits the situation being considered. You can do this on the board in 2 easy steps.

i. ***ID decision points chronologically***.

Ask what determines the lover's fate. When the decisions of the tiger-keeper (TK), the princess (P) and the lover (L) are identified, you need to agree upon an order in which to note them. From repeated experimentation, I've found that the order TK → P → L provides the most clarity and have presented this lesson and its supporting materials using this sequence. (It can be considered a chronological order of decision making because the princess can always change her intention *after* receiving the tiger-keeper's information and the lover must make his decision *after* receiving the signal from the princess. You can use any order you deem best, so long as you stick with it and are consistent.)

Draw 3 empty boxes about as big as your fist near the top of the board and write "TK" above the leftmost, "P" above the center box and "L" above the right hand box. If helpful, you can tell the class that this is our "Code Box" for untangling the story.

TK	P	L

ii. Record decisions of truth or trust as T, falsehood or mistrust as F.

Return to the 1st version of the story that you shared aloud, rereading it if necessary. Ask the class whether the TK told the truth or lied to the princess. Agree upon a pair of symbols for recording his decision. Convenient symbols are "T" to mean decisions based on truth or trust and "F" for those based on falsehood, lies, misdirection or mistrust, but let the class choose the pair of symbols which works best for them. Record the tiger-keeper's decision in the box below his initials and then do the same for the princess's decision and then the lover's decision. (In the example at right, the tiger-keeper tells the princess the truth, she tries to save her lover but he believes she is trying to kill him and chooses the other door.) Tell the learners *not* to copy down these notes as you will rewrite them neatly soon.

TK	P	L
T	T	F

Re-examine the 2nd version of the story shared aloud. Draw out a new Code Box and repeat the process of recording the choices made at the 3 decision points. Do this for as many versions as you read or have time to review. Be aware that while no 2 writers' compositions are identical, they may still resolve themselves into the same "version" of the story if the 2 writers have chosen identical decisions for each character.

Each combination of 3 symbols thus becomes a code for that particular version of the story. After you record the codes for the compositions you shared, learners should volunteer other possible 3 symbol codes while you write each quickly on the board, even if the suggested codes are repeats of ones already noted. When there are no more suggestions, say you want to rewrite these codes neatly where they can be saved. Ask the class to point out any duplicates and erase them so that only unique codes remain. (There should be 8.) Rewrite the remaining codes on a grid in the following order:

TK	P	L
F	F	F
F	F	T
F	T	F
F	T	T
T	F	F
T	F	T
T	T	F
T	T	T

iii. Record each 3 symbol code in its Base 2 position, lowest to highest.

For reasons that will become clear when you teach the fundamentals of binary mathematics, it is very useful for learners to have the version codes displayed in the specific order of increasing Base 2 value. (Don't worry if this makes no sense to you right now. Just understand that the more Ts a version code has on the left, the higher its value is.) You should end up with a list in this order, top to bottom.

iv. ***Name each version and identify whether it ends in marriage or death.***

Now comes the fun part. Since each code represents a particular set of decisions, we can give that version of the story a descriptive name based on the specific decisions made by the 3 characters. For instance, F F F might be titled "World of Lies" because no one tells the truth in that version. T T T could similarly be called "World of Truth". Along with a title for each version, the class should figure out and explain aloud whether the lover lives or dies as a result of those 3 decisions. You can record his fate in each version with a special symbol added to the right of the 3rd box. (You could use a ✓ for survival and an ✕ for death or perhaps a heart or skull.) The 3 examples below are for your understanding and should *not* be read to the class now. The individual styles learners use in reciting the versions and determining marriage or death is one of the most important and enjoyable elements of this process.

TK	P	L		
F	F	F		
F	**F**	**T**	✓	→
F	T	F		
F	**T**	**T**	✕	→
T	F	F		
T	F	T		
T	**T**	**F**	✕	→
T	T	T		

FFT: The Trusting Lover
The tiger-keeper lies to the princess who tries to use that false information to kill her lover. He opens the door she points to and feels his trust confirmed when the woman emerges.

FTT: The Jealous Tiger Keeper
The tiger-keeper lies to the princess who tries to save her lover. He trusts her but she watches in horror as the tiger leaps from the door she indicated.

TTF: The False Lover Punished
The tiger-keeper can't bear to lie to the princess who decides to save her lover. The lover just can't believe she'd send him to the arms of another, so he opens the opposite door and pays the price for his faithlessness when the tiger leaps out.

v. How can you tell we're done? (***You can't. We need another method to prove that.***)

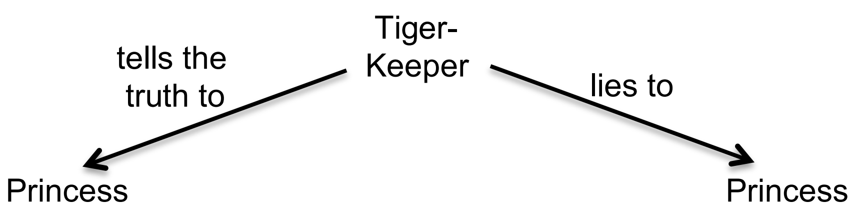
The problem with the brute force method is that it requires a 2nd method to prove that there are no more possibilities available. Learners may insist that they can "tell" that there are no more 3 symbol combinations possible, but ask them how they could be sure if there were 4, 5, 6 or 39 decision points to consider.

B] Tree Diagram:

1 tiger-keeper decision pt → 2 princess decision pts → 4 lover decision pts → 8 end pts

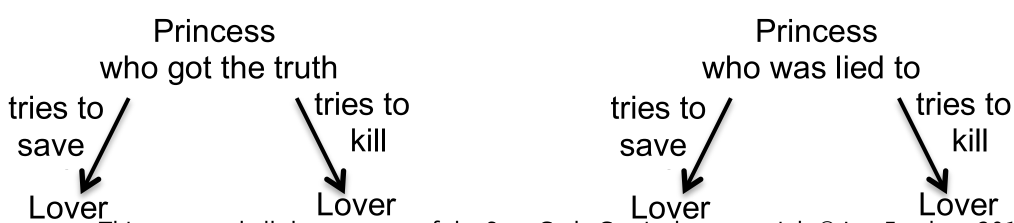
Of our 3 methods, the tree diagram is perhaps the most universally effective for understanding how the decision of 1 character sets the story down a particular "branch" of possible plots and how all 3 decisions generate an exact path through all possible branches of the story.

The diagram starts with a single point labeled "tiger-keeper" (unless you decided upon a different order for the decision points) and shows that the story can branch in 2 possible ways based on whether he tells the truth or lies to the princess.

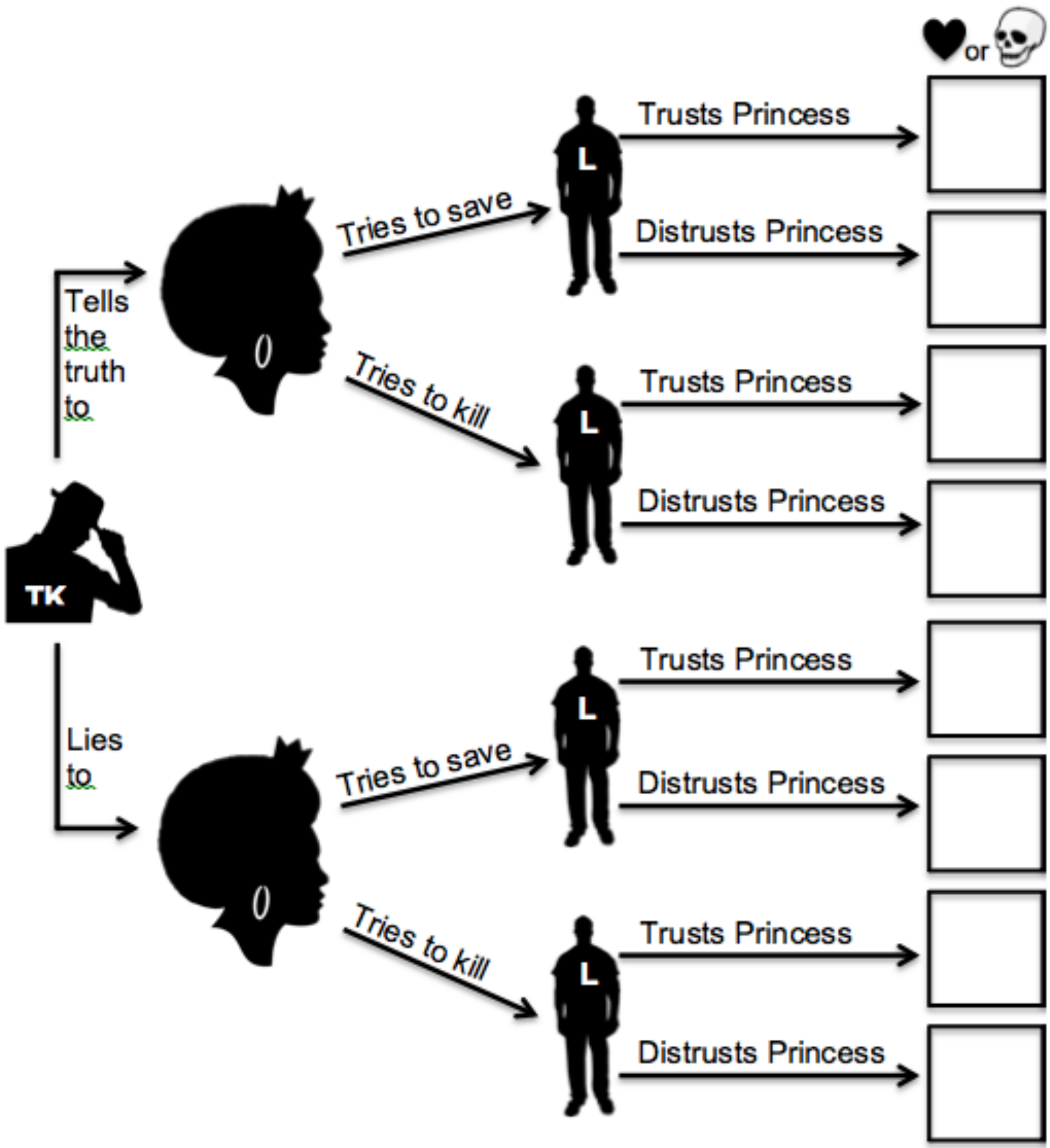


This creates 2 decision points labeled "princess", but it's crucial to understand that the princess does not make 2 decisions. The tiger-keeper's decision point essentially splits the story into 2 possible realities, 1 in which he has lied and 1 in which he has told the truth. These realities cannot exist in the same version of the story because he must either tell the truth or lie, not both. The tree diagram shows every possible version simultaneously, but each version is a separate reality from the others.

The princess must now decide whether to truthfully point her lover towards the door she believes will save his life or to deceitfully direct him to his death. This choice is represented once for the reality in which the tiger-keeper has told her the truth and once for the reality in which he has lied.



Her decision points thus produce 4 possible realities in each of which the lover must decide whether or not he trusts her and to open the door she is indicating. The results of this 3rd decision point (the lover's) generate our 8 possible versions of "The Lady or the Tiger?", each with an outcome of either marriage or death for the lover.



A complete tree diagram of the 3 decision points in StoryCode's "The Lady or the Tiger?", read left to right

Learners are often quite experienced with tree diagrams from their math studies and a volunteer could draw out the diagram on the board in their own style, perhaps with input from their colleagues.

A Note about "Tree" Diagrams
Tree diagrams (or, more specifically in this case, "decision trees") are wonderful analytical tools that can be used in all disciplines and many phases of life in general. They can be drawn in different ways which usually only reflect the aesthetics of the person drawing them. Many people start from a central point at the top of a page and work downwards, and although the resulting diagrams look more like mountains than trees, no one ever refers to them as "mountain diagrams". The diagram above proceeds from a single point on the left to 8 final branches on the right, giving more of a look of a fallen tree. The only way the diagram will actually look like a standing tree (or at least a bush) is if you begin in the center of the bottom of a page and proceed upwards, but many people have an inborn resistance to looking for starting points at the foot of a page. The diagram makes sense no matter where you begin, so please choose a format that works best for your learners, perhaps in consultation with colleagues who have worked with them on tree diagrams previously.

Each learner should copy out the complete tree diagram in their notes in whatever format works best for them. One of the digital files for this lesson, *LorTchart.pdf*, includes the above diagram with additional organizational notations. You may want to distribute the chart (or one of your own design) if you wish to scaffold learners' tree diagram skills. Distribute such a chart only *after* your students attempt their own tree diagrams.

C] Function: permutations and exponential growth

The expanding possibilities created by our 3 decision points can, of course, be calculated and represented as a mathematical progression or equation, a fairly simple one for those familiar with exponents and permutations. The number of possible versions depends on how many decision points there are (3) and how many options are available at each decision point (2).

i. ***How does the number of True/False choices affect the number of possible versions?***

Basically, when one of our 3 characters gets to choose between their 2 options, they double the number of versions of the story that are possible. There's only 1 version of the story until the tiger-keeper gets to lie or tell the truth to the princess. After that, there are 2 versions possible, 1 in which he's told the truth and 1 in which he's lied. When the princess chooses whether to point her lover towards life or death, she doubles the 2 possible versions to 4. The lover's final decision of whether or not to trust the princess doubles these 4 versions to our final 8.

What's essential to understand mathematically here, however, is that this doubling does NOT happen because we are multiplying the number of decision points (3) by the options available at each decision point (2). If that were the case, there would be only 6 versions of the story because 3 multiplied by 2 gives us 6. What IS happening is that the number of options available to each character (2) is being raised to a power, i.e. increased exponentially. We have 8 versions because 2 is raised to the 3rd power since there are 3 decision points: $2^3 = 2 \text{ X } 2 \text{ X } 2 = 8$.

ii.a. That is to say:

$$8 = 2^3$$

the number of decision points

the number of versions

the number of options per decision point

If we introduced a 4th T or F decision point to the story, there would be 16 possible versions because $2^4 = 2 \text{ X } 2 \text{ X } 2 \text{ X } 2 = 16$. This is a form of permutation, a mathematical way of identifying sets of items in which the *order* of the elements in the sets matters. For instance, amongst our 8 versions, the set TFF represents a different story ("Unreliable Lovers") than the set FTF ("Unreliable Men"), even though both codes have one T and 2 Fs. The order of those Ts and Fs makes a difference in the story.

b. Written as a function, the formula looks like this: $f(x) = a^x$

In math, a "function" can be thought of as something you do to a number that produces 1 (and only 1) result. A function is often (but not always) symbolized by a cool, script, lower case F: f.

Let's consider a function that adds 2 to any number we start with. This function could be written as $f(y) = 2 + y$. In plain words, this is saying "Take any number (y), hit it with our function (f) and the result is 2 plus whatever number we started with." (It doesn't matter if you use x or y or most any other letter inside the parentheses of the function.)

The result of a function can be almost anything, even the same number you started with, like when you multiply a number by 1. As a function, multiplying by 1 could be written like this: $f(y) = 1 \text{ X } y$. Even more simply, it could be written as $f(y) = y$. The function for multiplying by 0 can be written as $f(y) = 0$, since any number multiplied by 0 is 0.

Because functions show how things work in a very universal way, it's worth it to rewrite our formula for the number of versions as a function. Our function must tell us the number of versions possible based on how many decision points we have and how many options there are at each decision point. The function must be accurate for any number of decision points. We can write it like this:

function for finding # of versions with x decision pts $\longrightarrow$ $f(x) = a^x$

the number of decision points

the number of options per decision point

2) What if the lady and the tiger could secretly switch places on their own?

Suppose the lady and the tiger could secretly and by their own choice switch places without anyone knowing. That could be seen as introducing into the story a 4^{th} decision point with 2 options. How many versions would be possible now and how can we show that?

Each of our 3 methods produces the expected answer, 16, with our function doing it most quickly: $f(4) = 2^4 = 2 \times 2 \times 2 \times 2 = 16$.

Using a tree diagram, you would have to add 2 branches to each of our 8 previous "end" branches and when you counted up the new end points, you'd get 16.

Brute force starts to become a hassle with 4 decision points. There's quite a bit of labor involved with writing out all 16 permutations of 4 T or F characters and making sure you haven't repeated any:

FFFF FFFT FFTF FFTT FTFF FTFT FTTF FTTT
TFFF TFFT TFTF TFTT TTFF TTFT TTTF TTTT

Again, brute force also requires a 2^{nd} method to prove that you're done. Any learner predisposed to using this method should be starting to see the power and convenience of using either the tree diagram or the function instead.

Mini-Lesson 1: What do the decision points of "LorT" have to do with computers?

By this point in the lesson, learners may have begun demanding to know what this all has to do with computers. Now we make those connections.

1) Computers work with information in the form of 1s and 0s.

Some insightful learner may suggest this concept, but it is much easier to say than to explain. For this lesson, what you need to understand is that all information used or stored by a computer ultimately boils down to long sequences of 1s and 0s. If we could speak about a computer's view of the world, we would have to say that it starts by knowing only 2 things, a 1 and a 0. A computer knows that a 1 is completely different from a 0.

A] 1s and 0s can mean almost anything.

From only these 2 elements, computers build up amazingly complex codes that produce every color we've ever seen in a digital photo, every note we've heard on a CD or iTunes, every flying bullet or laser blast in every video game ever.

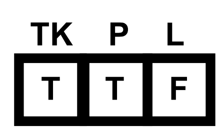
B] The meanings of all these 1s and 0s are determined by humans.

Computers didn't figure out how to do this by themselves. Digital machines are incredibly fast and accurate at sorting through these codes, but it is people who had to make up the codes. Actual humans have to figure out what all the 1s and 0s mean and how to put them all together.

2) Using 1s & 0s requires a "protocol", a method or order of doing something.

I first heard the word protocol in some movie, probably a military or legal thriller, "You're not following protocol!" or some such line. When a group agrees upon a single method of doing something, they have established a protocol. Protocols often break a task into steps and specify the order in which the steps are to be done. There are protocols for going to the toilet during school, for washing your hands in a hospital, for setting tables in restaurants. The military has protocols for everything from shining boots to launching nuclear missiles. Turning 1s and 0s into useful information requires many complex and powerful protocols, but we can begin with the very simplest ones.

A] Define the protocol we used to code "The Lady or the Tiger?"



When we organized our versions of "The Lady or the Tiger?" using our Code Box, we actually created a very useful protocol. What are the parts of that protocol?

i. symbolic meaning: ***T = truth or trust, F = falseness***

The first part of our protocol defines a set of symbols. Our set is small. There are only 2: a T for trust or truthfulness and an F for lies or mistrust.

ii. positional meaning: ***TK → P → L***

The second part of our protocol controls where those symbols are used. Remember how the codes TFF and FTF stand for different versions of the story even though they both have 1 T and 2 Fs? That's because our protocol gives different meanings to the 1st, 2nd and final symbol in each 3 letter code: the first symbol represents only the tiger-keeper's decision, the second is always the princess's and the last must be the lover's. It's not just the symbol that creates the meaning. It's also where in our code the symbol occurs.

B] The word "binary" describes a system with only 2 parts, T or F in our case.

"Bi" means 2. *Bi*peds have 2 feet. A *bi*cycle has 2 wheels. A *bi*nary star system has 2 suns, 1 orbiting around the other. Because computers must see the world as being made up of combinations of only 1s and 0s, we can say that they binary machines, that they use binary information. Used by itself as a noun, "binary" has come to mean the nearly incomprehensible (to humans) strings of 1s and 0s used and produced by computers on their deepest, most mechanical levels.

C] "binary protocol" = ***a method or order of doing something using only 2 elements***

Since our LorT Code Box protocol uses only 2 elements, T or F, we can accurately describe it as a "binary protocol", a method or order of doing something using only 2 elements.

Some learners might object that we actually use additional elements because we have the ✓/✕ or heart/skull system for recording the lover's fate after the 3 decision points. Those extra symbols are not really a part of our protocol, however. They are results we can foretell from reading the 3 actual parts of our protocol. (If this is unconvincing, you can always replace the extra symbols with T and F, T standing for marriage in the 4th position and F standing for death. Voila: a pure binary protocol.)

3) Switch the Ts and Fs in the Code Box to 1s and 0s.

So how would a computer use our protocol if all it knows are 1s and 0s, not truth or falsehood?

A] 1st identify the meanings of our symbols: T = True = 1, F = False = 0

Well, it turns out computers actually do know something about this. It's standard computer practice to have a 1 stand for "true" and a 0 stand for "false". Switch all the Ts on your brute force chart to 1s and change all the Fs to 0s. Suddenly, we have a set of codes that are right at home inside a computer's binary protocols.

B] Identify the meanings of the positions of our symbols: 1st # = TK, 2nd = P, 3rd = L

A computer doesn't understand or care about what all these 1s and 0s really mean to us humans. Its job is just to keep them organized. As long as we understand the significance of each symbol's position, the binary protocol works fine.

4) Do we need to record the ending with another digit?

A] ***No. We can always figure the ending out from the 1st 3 digits.***

As mentioned in 2.C above, the 4th symbol (or digit) symbolizing the lover's fate is not really necessary because we can always determine it from the 3 original digits. We do this by talking through each version and figuring out what came through the door, the lady or the tiger. With the brute force chart and the tree diagram on the board, however, some shortcuts for determining the lover's fate should become apparent.

B] Can you see any patterns for the endings?

TK	P	L	
0	0	0	×
0	0	1	✓
0	1	0	✓
0	1	1	×
1	0	0	✓
1	0	1	×
1	1	0	×
1	1	1	✓

A single 0 in the Code Box results in the death of the lover.

Check it out: any time only 1 character misleads or mistrusts another (011, 101, 110), the tiger leaps from the door and the lover dies. An even more accurate observation includes the first code: any time there is an odd number of 0s, the lover dies. Learners often describe this pattern more interpersonally: in the eighth code (111), all the characters act truthfully and trustingly and the lover lives. If 1 person, any person, acts deceitfully, this "World of Truth" is altered and he dies. If 2 people act from deceit, their lacks of trust/truth cancel each other out and the lover lives. If all 3 behave deceitfully, then 2 lies cancel each other out leaving the third lie to doom the lover.

5) TK-P-L Code Box = a binary protocol

So, we have now coded and analyzed every possible version of "The Lady or the Tiger?" using a binary protocol. Computers use an incredible number of such binary protocols, all designed by humans, to do everything they do. What we've accomplished in this class may not seem like a huge deal, but we've laid the groundwork for many technical understandings to come.

Time left in class? Mini-Lesson 2: ASCII, another Binary Protocol

You might have guessed by now that a computer, seeing the world as it does as combinations of 1s and 0s, cannot understand the letters of the alphabet in the same way that so many humans do. After all, there are far more letters than T and F.

1) What happens inside a computer when you press a key while typing?

Pressing a key on your keyboard can make all sorts of things happen, so let's be specific about what we're talking about now. Let's say you're writing an essay in a word processing program or sending an email. The text that's appearing on the screen is not triggering any operations in the computer, unlike when you hit Ctrl + P or Apple + P to print. The letters are just appearing on the screen and being stored in your computer somewhere. How is this possible if the computer only understands 1s and 0s?

2) The computer has a code of 1s & 0s for that character but shows the letter on screen.

Pressing the keyboard signals a particular code of 1s and 0s inside the computer, a code that stand for that specific letter. To display it for you, the computer matches those 1s and 0s to a certain pattern of color on the screen. That pattern of color, usually black against white, is the letter we see.

3) Only computer users see this information as anything other than 1s & 0s.

If we didn't need to see the letters we type, there would be no need for the computer to form these color patterns on our screen. Our writing would just sit inside the computer as 1s and 0s. Because computers are tools we invented for our use, we almost always make them convert their 1s and 0s back into things we can understand more easily like letters, colors, sounds, pictures, music and video. When computers communicate with each other, they don't bother with these conversions and pass everything back and forth as 1s and 0s.

4) Hand out the chart of ASCII binary codes.

Feel free to use the ascii.pdf handout included with these materials.

ASCII stands for American Standard Code for Information Interchange. Work on this protocol started in 1960 and it became the world's most used **binary protocol for the alphabet** until 2007 when it was surpassed by a more efficient but more complex system called UTF-8. ASCII is a great model for understanding binary protocols. It uses a code of 8 1s and 0s to stand for each letter in the English alphabet, plus codes for numbers and things like blank spaces and dashes. As the handout shows, ASCII has separate codes for capital letters (codes beginning 0100 or 0101) and lower case letters (beginning 0110 or 0111) since upper and lower case letters make a difference to us.

5) Has anyone ever heard of a simpler binary protocol for the alphabet?

Morse Code, *invented 1836*:

> The American painter Samuel Morse was one of the inventors of the telegraph, a system for sending information over a metal wire in the form of electrical pulses. A protocol was needed for these signals to be understood and Morse came up with a code for the alphabet made up of long electrical pulses and short electrical pulses. (SOS, the standard emergency signal, is 3 short pulses, 3 long pulses and then another 3 short pulses.) Morse's protocol is also easy to use without electricity, letting you spell things with flashes of light or blasts on a horn. It was immensely important for communication throughout the 19th and 20th centuries and is still in use today. Movies feature Morse Code transmission in many different forms: telegraph operators clicking away on devices that look like staplers, stock traders reading lengths of "ticker tape" unspooling from machines, wilderness patrols using mirrors to signal between mountain tops with reflected light, trapped miners banging out codes on pipes.

Morse Code is ingenious, but it is not a binary protocol. Not all the letters in its alphabet are coded like S and O, as 3 long or short pulses. Some letters are symbolized by only 2 pulses. For the 2 most common letters in English, E is coded as a single short pulse and T as a single long pulse. This allows Morse Code messages to be shorter than ones in ASCII, but it also means there must be some signal that indicates when a letter is complete, since it might be 1, 2 or 3 pulses long. When a human learns to tap, flash or sound out Morse Code, this extra signal is a silence that separates the pulses of each letter. It gives a rhythm to the code and becomes part of the transmitter's recognizable Morse Code style.

Perhaps because humans are so good with rhythm, Morse Code's trinary protocol was conceived before ASCII's binary one.

HW: Write your name in ASCII.

Though StoryCode avoids busy work, at first glance this assignment seems rather silly. What does anyone really learn from using an ASCII chart to translate their name into 1s and 0s, especially if your name is Francisco or Ernestine? There is a benefit, more gut level than brain level, but it is worthwhile and only requires coding one's given name.

In completing this homework, many learners will commit the cardinal sin of introducing a third symbol into their binary code, making it no longer binary in the same way Morse Code is not binary. Consider even a short name like Ben. Just looking at it in binary can be aggravating: 010000100110010101101110. It is human nature to organize annoying masses of information. Written out as above, the string of 1s and 0s communicates almost nothing to us. Divide the string into its 3 component letters, however, and we can see some patterns in the 1s and 0s. Following are 5 common ways of breaking up the string:

A] 01000010-01100101-01101110
B] 01000010,01100101,01101110
C] 01000010/01100101/01101110
D] 01000010 01100101 01101110

E] 01000010
01100101
01101110

Every one of the transcriptions above uses a third element and is, technically, inadmissible as binary. The first 3 introduce the dash, comma and slash characters, respectively, while [D] makes use of the blank space as a third, organizing symbol. [E] is the most dramatic, using an entire new line as its third element. Learners may groan when this error is pointed out, but it leads them to an important appreciation about computers: they require no organizing symbol. They may know only 1s and 0s and they may have a robotic rhythm, but they are incredibly good and fast at working with those 1s and 0s, counting them and keeping them straight.

In addition, writing out all these 1s and 0s, if only once, gives a good appreciation for just how many of them go into even the briefest email. How many 1s and 0s are there in a 3 page essay? Perhaps that's a good follow up homework assignment.