StreamBox

[Captioner standing by]

>> PROFESSOR: I see this is the microphone. You can tell but I have a - - try, when he has. Then we can talk. Good? Okay so we are going to make sure that we touch on the key points so that you don't mess. I am looking here at lecture one. Slide number seven. These are based on the questions we had from people.

>> SPEAKER: Wait, wait.

>> PROFESSOR: We will have the group interacting with the bot called sim allow. I will show you the questions. And in this bot the answers will be accumulated. So, suppose he answers something. Over there he answers something. They are in the same group. Then the group is going to accumulate those points. And those would be bonus points is that clear? So this will allow us to have interactions individually and used continuously to the group selection of points. Those points will be bonus points from the group. Any questions about that? Today will be a very simple exercise on that. And as we move on into the rest of the material, this interaction with the bot will be more interesting. At this point we want to make sure everybody knows where to go in that bot, how to answer questions in the chat so that your group gets the benefit. Is it clear? Okay, we will see how this goes in the second part of the graph. I decided today to go through, certain aspects that from my viewpoint, because there are difficulties with a language like Python, it is difficult or anything like that but logically, when you use other languages you will find there are better design languages. But you have to lead with this somehow.

>> STUDENT: I just picked up this class and - - [away from mic]

>> PROFESSOR: This is not the right time to ask the question. It has to be online. So division, this kind of stuff one backslash to backslash. I may explain lots of times giving this here. That is why you really need to go to this participation activity, you will see the effects of these things. Understanding it is two and one backslash. These participation activities are not required for you to do. But if you do them you will get bonus points. The idea of participation activities is they are designed to help you. So are the homework problems. You really want to get the benefit, of the stations, that is the point. Your TA's have been instructed that in the - - the point is for you to solve most of the homework problems. That is the point. When you have to try participation before that then you will have more difficulty with that. Now, according to the questions for the class, we notice this is a very diverse group of people. Divers in the sense of the background knowledge that we have there are certain things that might be elementary to some of you. Those of you that feel like you are a Python expert already. We have the possibility and you have to decide this

yourself. To replace some of these homework problems for these people that want to do it. This is completely optional. To replace them with some other more complex assignment. You think I'm a Python expert, I really want to go through these details. Then you will have the opportunity. He will replace homework three prior advanced homework three. So you don't have to do homework three. You have to do the advanced homework. Is that clear? How many of you, I would like to see a show of hands. How many of you who like to do the more advanced stuff?

Excellent. Talk to us and we will do that. This model is another item here in the lecture. Can anyone tell me in one phrase what is this?

>> STUDENT: Remainder.

>> PROFESSOR: Remainder? What about you? The remainder. Any other? Let me show you a picture. The idea of this picture is that there are two segments. This is one and this is another segment here. And this is another number. This is number one and this is number two.

The way I draw these pictures at this point is assume that this here represents the number zero. I am going this way. This is a positive number. Therefore going to the left, this is a negative number. What is the purpose of this operation? It's to take this number, in this case positive. Divided by this number. In this case negative and get the remainder. That means that what you are doing is taking this number, this interval and you are dividing to the right until you pass that extreme. And that will give you the, what is this? The remainder. 16. -4. Were going to write this segment how many times until you pass here? Four times. When you pass here you get to 16. The difference is the remainder. Raise your hand if you got this? Okay, some of you have to practice if you haven't seen this. If you go the other way around I could have 15. And I could have the second member here. So in this case those zero, the number and the second number here you will write to the left. Until you pass here and that is the remainder.

>> STUDENT: Writing it to make a little section and seeing how many sections go for the big line until it overflows.

>> PROFESSOR: The overflow is the remainder. Yeah? This has a name, models. You can look at the example. This is the symbol and it's important to understand why the answer here is one and why the answer here is -1. That is for you to try. That is also with real numbers. Something I would like to point out is this business of extremes and numbers. In my experience this has been a little bit cumbersome. This is one of those things in the

language
that I like a lot but you are going to use it. It is important. So follow the rules of the language even if you don't like it. This is one of those things. Extremes and numbers. extremes and numbers. What are extremes? Sequence of characters. And numbers I hope I don't
have to explain. The question is how you go back and forth between distance. You have extreme. How can you get a number out of it? You have a number, how can you get extreme out
of it? There are certain things here that are a little bit confusing. Let's look at this example. Remember that in Python, I hope you know this now. You can use single, double or
triple quotes. This is a way to say extreme. There are different ways to explain the same thing. When you can say these are extremes, okay. If you want to put something like this inside an extreme. Because this is between quotes. Inside double extremes to bring this and
will get exactly what's inside the double quotes. So you can look at this. There are multi-extremes. Sometimes you would like to get something that looks like this. How do you
do that? Three single quotes. So you put three. Single quotes then, this is what you get. This one over here, this over here. And this next one over here. To express that in this language you use triple quotes. Now, these are special characters. Because this is a sequence of characters. There are certain special characters that you need to be aware of. Usually these are special characters that have something like this. In front of them. So you have something like this and say this is something special coming. The most common one is
this one. What is that? New line. That indicates in this sequence there is a special thing called new line, new line characters. What is the other one? Backslash tea. What is that? Anytime you have value want to insert a copy and use that. For example, the way to interpret
this, this is a string. The first part is ABC, new line DEF G, 2011. Do that and you get exactly that. Use triple quotes for multiple strings. This business of going back and forth between numerical values, look at this example here. This is saying quotes are extreme. Put
that in parentheses with INT, what this is saying is you are going to create this extreme. As an interval. So this is not to five anymore when you do that. I want to emphasize that. This is, when you have quote 25 quote, that is extreme with two characters. Two and number
five. That has no meaning except to characters. When you put INT in France, that is a completely different object now. This is saying that this extreme is going to be interpreted now as the number as you are use to 25. Raise your hand if this is completely clear. I see some holes. You have this understanding that, although this exercise and you will see when

you do this, INT, this is an integer. You can find all this is really an integer, yes. Type and it tells you this is an integer. What about quotes? you have quotes floats that means it is extreme now. It will be interpreted. The extreme minus 25.5 which is a collection of five characters. That extreme is going to be interpreted as a floating-point number. By putting a float in front of that. If you don't know what that means, print it and ask for the type of it. And you will see this is -25.5. I want to emphasize that visually they look the same but they are not. Quotes to 5.5 is not the same as -25.5. -25.5 Is a value. A number as we humans interpret that. And you can go back and forth. So this is important to be aware of. You can look at where, if you don't do this property then it tells you this is an error and you can check why it is. What about this kind of stuff? Now this is the other way around. You take the number three. You start with variable A. Here you say STIR parentheses A. What
is that, anybody? Back there. So a is now extreme. What is three? The question is what is three? For you, we have this and after that you have as TRA. The question I am asking is what is this? Versus that. Yeah.
>> STUDENT: It's a character.
>> PROFESSOR: So as TRA is what? Which one in this case? Here, don't forget I am first using
an assignment statement. The integer three is being assigned to variable A. If after that I say STRA I am asking what is this object, STRA. What is it? Extreme but in this complete case what is it?
>> STUDENT: A string of terms.
>> PROFESSOR: A is a variable that contains the integer three. As the value of a. If after the, we have STRA what the hack is that? Yeah. what is your name? Say that loud. Is the value of the variable A converted to extreme object. So the value or the variable A is now an
object. That object is extreme. Raise your hand if everyone got that beautiful answer. Okay. This is important because when you are running code in this language, at some point you
are going to see error. And the question is, why? When you look at this and say it's the same thing, that language. At that point you have to ask yourself what is the variable, what is the value of the variable and what is the type? So the way to understand this is here. When you are here, a is a variable of what type? Integer. At this point a is a variable of type integer with a value three. Three things. Type, value. I'm talking about the identity of that variable. That identity the location where that is stored. Usually you don't play with that but it is good to be aware of. Here this is not an integer anymore. The type of this is? Extreme. So far so good? Okay. Now what happened here? What is this IP ART after this statement? Yeah? An integer. What is the type of this? It's an integer. What are we doing here now? We are casting. We are changing this integer into an extreme. We are transforming these into extremes. This is the transformation here with as TRA. Holy guacamole. This is not an integer anymore, this is extreme. This is not an integer anymore. This is extreme. And we have here is something in between. An operation between

extremes.
What is the operation between extremes? We use A+. Yeah, for all of those experts. This combination, you have extremes, you put plus and you are not adding really but this silly language is the way the interpreter takes thought. Concatenated. These are completely clear.
So to understand the operations you need to understand the arguments and what is the type of
parameters you are using. So now when you get input which is something like this, it is simple. The parameter, the subject is a string. If something is tight. Then that's according to this particular variable. Suppose that guy is allowed or not is what you would say. I want to now show you something. This one, this funny inconsistency in this language you have to be aware of. Input. This is the extreme. You are asking this in the consort. Whatever the user types are after that in the console is going to be a store called NUM one. So when you type in this and it's executed, you use it by five, this is what you are getting. Five. Enter another integer. Nine. Oh, okay. You enter five and nine. This is saying this is a problem. This is taking integer five and nine. The sequence five and nine. Taking those two numbers. We would like to multiply them. We would like to print the product of number one and two. It is completely logical to me that that should be this Times that. It is completely logical. But Python does not like it. Why? Yes? What? He says that he doesn't like, Python does not like that because?
>> STUDENT: When it says integer.
>> PROFESSOR: So NUM one, this is what? This is a string. This MUM2 is? Extreme. We are multiplying two extremes. Yes? Which is logical for the human point of view because you said
come on, this is integer five, this is integer nine. I am multiplying them. Why are you giving me a hassle? Well, for the language you don't have two integers. For the language, Python interpreter, research extremes. Two extremes cannot be multiplied in Python. God is
the moral of the story. Extremes cannot be multiplied in Python. Yeah but we just saw an example that was NUM 1+ NUM two. But we don't have extremes there. We are concatenating
these. So extremes can be concatenated. They cannot be multiplied. How do you solve our problem in putting integer, and putting integer? You transform the extreme into an integer by
doing this. Now you have two integers and you are happy. Is this completely clear? Before executing an operation you must be sure that the parameters are correct. And you are going to
multiply two numbers, that is the intention here. The user is typing this. Typing something better extremes. Before you do integer operation have to put in the corresponding team because then you can multiply integers. It happens to be that this, why did I say - -? Elementary school kids. This person enters here 3.14. It's elementary school. You type in integer and types 3.14. What is the problem? 3.14 is not an integer. But because the code

is correct, this piece of code is perfectly fine now. So the problem is not with the code. The problem is that the user entered something that he was not supposed to enter. There is nothing wrong with the code. So this is a type of error that you must be aware of. I will not type in error but it is very face of actually, this business of dividing by zero. You took precalculus, calculus, trigonometry, whatever. You cannot divide by zero. In Python you cannot divide by zero either. A very common mistake, anytime you divide by something, make sure the denominator is not zero. This is a type of error. There are participation activities that would allow you to do that. Those were the main issues I wanted to point out about lecture one. Lecture two, we just mentioned it is important in the parameters in this language for you to be aware of the parameters and that they are correct. If they are not correct you have to use some form of casting. The importance is that the objects have identity or type and value and those are the things you have to be aware of. Using an example here, a is an integer. Here it is saying print IDA. That means the place in which this particular variable a is story memory. Usually the answer is some long number that has no meaning. No, you took this a, you had one value and print the idea of a. And now this is in a different memory location. Do you need to be aware of this? Maybe for the big programs this could be useful. Okay. Expressions you can combine. If you use an - - you can combine them. Simple functions. Remember the last time we mentioned the definition of functions. You can define your own. For example double of X, that is just X supplied by two. The purpose here is you can define something more complicated as a function of X, write a more complicated function. At the time you don't have to write the complicated function, you can use the name of the function. And you can use the exercise for this. Products, functions with more than one parameter. For example, here you have a function with three parameters. X, Y, Z. You are defining this way, you can put whatever you wish. These are the three parameters and is saying I want to encapsulate in this name the product of this three input parameters. When things start becoming large it is good to put things into a special collection. To access special collections, the model, you already explained the syntax of this. Mrs. the library. Think of this as a library and we talk about models. This function could be W of XY. Lynn for example, 33 is 27. You can change parameters. This particular one you may not be aware of. But here you are taking this exponent,.5. That seems to be a problem sometimes. What is that? What is this? These people that said they would advance, I want advance people to

give
me an answer. Where are they? Yeah. This is, very good. This is what you are used to. Write this, write that. Two. Why this? Because it's very useful for many computations. Is that okay? Two X to the .3? Yes. The point is this is just a special case that you can specify the perimeter there. Something more interesting behind these things. Maybe I do it now. What is this? This is clear now. What about this? Two raised, sorry. Another one. This one isbetter. -1 Raised 2.5. What is that? What about the other people? Yeah. Two indicators. Is that okay? Before somebody tells us .5 translates in human language as a square root of -1. I am just doing, literally, anytime you see this that can be replaced by the similar, mathematically speaking and this is this. The square root is that. -1. Come on, guys. You claim this is to Elementary of a class. Tell me something. No other number? What is this, guys? Square root -1. Yeah? Is this an imaginary number? That is what it is. But that is not the right way to think about it. From the programming language point of view. What do I mean by that? It is important to understand what is the meaning? The meaning. Oh, this. What is this symbol? The meaning S, whatever this is it is equal to this, Y. If and only if X is equal to Y squared. This is the definition. Just a symbol. The square root of X is Y equal to Y squared. What is the square root of -1? Something equal to Y. If only, come on guys. Come on, come on. If negative one is equal to Y squared. Here, when I ask you to compute square root of -1 you have to give me a number. So that when you
raise it to the second power it gives you negative one as an answer. Is there a real number that satisfies that when you square at it is equal to -1? No, there is no real number. If I ask you to compute this there is a way out. You say I cannot give you a real number. No real
number satisfies this. However there is all kinds of numbers. Invented by mathematicians that can scale this problem. That kind of number is what is called an imaginary number. This
one. This becomes what is called R. There is no real number to satisfy that. What's the point of this? The point is when you are using assumptions this is dealing with real numbers.
Therefore you put square root of -1. What is going to happen? What's Python going to say? There's a problem. There is an error. However, you are using X appreciation, you can take a number, raise it to the squared power and if that is -1 there is a way to compute with that. Conclusion, when you are using this mathematical libraries, it is important to understand what
the parameters are. The conditions of the parameters and what it is that is being produced. Here is the square root of -1. -1 Raised to the .5, square root of -1. It tells you there's an error. These are the conditions. If X is negative, Y is not an integer. Language will tell you. However there is another operation in the language called appreciation, that operation is to stars. You write 2-1 to the .5 that is this. This. That is that. This is -1 two stars.5. When you type that you get this. Why are there so many symbols? I NJ appear

in this kind of special use. This is not a complex number. So you can operate if you want, even in Python. With complex numbers. But separate yourself in terms of real computation and computation with complex numbers. The thing to be aware of, at some point you are dealing with this. You have to use not the usual operations but X appreciation with this caveat here. The functions, taking the setting of a number, taking the floor of a number. Can anyone tell me quickly, what is the setting of a number? The sitting of the number is going to the next integer. What is the floor? Going to the integer - -. What is G CD? Raters, divider. What is LCM? Least common multiple. When you write exp, or E raised to the power of X. This is special, EXP two when you have just done base two. Two ways to some power. You want to see what is two ways to the 35. The log function is provided in this library. You cannot specify the base because you can take log base E which is default or you can do here, this is log two. You can have imaginary functions and also important, this answer to this is in gradients. Important mathematical constants defining the model that you can use those names. MPI is Pi, the number pi is very important. What kind of number is pi? It is a - - number. What does that mean? It cannot be written down. With precision, with approximations. What about the number E? Also a rational number but these numbers are extremely important. Here are samples of how to use those functions. We spoke about random numbers in the last lecture. In the library, usually, if we want to be technically precise these numbers are called pseudorandom rather than random. Because a big question in mathematics and computer science, what is a random number? We are not going to enter into, this is very important. A big question, actually. But there are processes and algorithms. That we generate for you. A sequence of numbers. Which, statistically appear random. So it is statistical randomness. We call it pseudo-randomness. We mentioned this library. These libraries by the way, for random number generations are usually start with something called a seed number. The collection of numbers with a statistical properties, you can change the CD you want. Bottom line, there are mechanisms to generate numbers that are random in the library. For example, this is a function random in the library random in which we called out, print the number and this is a pseudorandom number. The important thing about this is these numbers are between zero and one. You are going to generate a collection of numbers that are pseudorandom. You can

specify a range and one of the most typical ranges are between zero and one. Because they
model what? What is a collection of numbers between zero and one that appear everywhere? No.
Something more important. What is a collection of numbers between zero and one? That appear
continuously. In data science. In computer science. Yeah.
>> STUDENT: Uniform distribution.
>> PROFESSOR: Probability distributions. What is a probability distribution? We are going to think about it forever. Remember this. Forever. Probability distribution you want to explain that to your brother or sister. It's a whole bunch of numbers between zero and one.
That add to what? The probability distribution is a collection of numbers between zero and one that add to what? In many processes you like to simulate probability distributions. To simulate them what do you do? Use which function? The collection of random numbers that add
up to one and it is nice because then you can use that as data this is an important function. How are we doing with time? Now, take the quiz? To close in the last 45 seconds of this part you can use this to simulate the control of I. You want to use a process of I, this is it,
the piece of code. Remember, with six faces you draw it. One, towards the leg. We were not going to sit down for one month because the way to simulate that is to use this random number
generator. So here is a piece of code. For games, also. I think that you can generate random rectangles for example. You can say random.uniform in uniform distribution between
three and four. That is generating a random width, now you can print out. You can - - and it will give you a random rectangles simulator. You want to know more about probability, you can
have something that follows what is called the - - distribution. That is using this particular function. Because you have very powerful libraries. And you can do a lot of stuff. You can generate, let's say a person's height and weight. How uniform you need to be between 1.5 and two because we are assuming we are not taller than the or smaller than four a
reason. The reason between 1500 pounds, you can create that randomly. Hide them away and now
print the two. Then you can go through what is called the body mass index. Body mass index,
everyone knows what body mass index is? The relationship between the way I am a little bit
wide and height. You can compute your own body mass index with that. When you go to the
doctor you say all right, your body mass index. You have to do something. To improve your

body mass index. This is a very used parameter to establish data on people and their physical
conditions. And you can simulate that. Okay. We are done with that. The quiz is open. It should be about what, 10 minutes?
>> STUDENT: I am having trouble connecting.
>> PROFESSOR: Sit down, we can help you. There is nothing I can do. Two in yesterday and today.
>> STUDENT: - - [away from mic]
>> PROFESSOR: Not the right time. If you have issues connecting come to the front.
[Students taking quiz]
>> PROFESSOR: After you finish the quiz stay because we are going to have the group interaction with bot. 10 more minutes. That's what you need to do after you finish the quiz. Log in. The quiz is over. Follow these steps. Follow the zoom link and at some point you will get to the sim alive. Step one, login through Canvas. Go to the syllabus, click lecture zero. Then I will give a length that you can click. You should get to sim alive. This shows you how It should look. That is the bot. It should look something like that. It should look something like that. When you get to sim alive you need to send a message. You are going to
identify yourself. The ID that you use, use your net ID and reverse it. Eventually when you get to sim alive you identify yourself. The student ID, reverse student ID. Are you there? What happened? Did you follow the steps? Are you in sim alive? So, yeah. You send a message. With the student ID like that. Did it work for you?