The

>>PROFESSOR: It should not take more than 30 minutes. So, we are going to give you a little piece of paper - okay, no charge for this, it is free, okay? And then we just are answering the questions and you record right here the question number and your answer. Each question has our number and it is written in here for you. Right? For example, this is group 31. Is group of 31 here? Yes? Group 31? All right. And the questions are P 20, P 21, blah blah blah.

After 35 minutes we will open the stream and then you start inputting and you commit to whatever. You start answering the answer, but if you don't write your answers here, you will not have enough time. Is that clear? Okay? After that, depending on how that goes then we will go to part two.

So, there's going to be an issue here, because - because - group 31 is here? Anybody from 31? Anybody from group 35? (POOR AUDIO) group 63? Group 67? Group 71? Group 75? Group 79? Group 32? Group 32? Group 36? 40? 40? Group 40? Group 44? Do not worry, we will get to you. Group 48? Group 48? Group 52? Group 52? Group 52? Group 56? Do not worry - which one, 60? You guys sit together, it will help you. Okay, I'm sorry, I'm sorry, I'm sorry. (POOR AUDIO) group 68? Group 72. 68? You are 68? (POOR AUDIO) 65? 69? 73? 77? 74? 74? 38? 42? You two guys? 42? 46? So, the idea is for the real midterm you know where to sit. So, next time we do not have to spend time doing this. 50. 54. 54? 58? 58? (POOR AUDIO) so, those are things that you need to do to create one of these blueprints and the first thing you need to do is need to define it and a way to define is very simple. So, here you use the keyword CLAS for class and you specify the name, whatever you want. Whatever name you want. And now you are going to initialize that here. Remember, you are finding a map for data. This is what this is doing. Oh, now the tricky thing that you need to be aware of is that there is a special method called INIT that stands for initialization and there is a under score and this method is very important. Anytime you have a name that has a_before and a_after, these are very special methods and they are called (NAME) methods and we will learn more about this throughout the semester and so what is this doing here? Well, this is initializing a class called time and Python uses this special parameter SCLF (SP?) and you can use whatever you want, but you know when I say class time I'm going to initialize myself. This is what this is.

Now, the way to think about this is that this is a collection of variables and this flexion of variables have values. So, with this particular thing is saying is this CLAS I am defining call time is going to have at this -- is going to have two variables and there could be a point of confusion here which is what is the name for the class? Is it time or is it self? The name for the class? It is time. The name is what you were defining about, it is time. So, this business of self is a trick and it is a trick and Python. It is a very powerful truck and it is saying we're going to initialize this graph and this very powerful treatment and remember these can be very complicated in general! And this is what allows Python to compare different very complicated object, because the objects are not going to be compared little by little, no,

this will object has a string and Python deals with this when it is going to compare.
That is the magic of it. Okay, so this is just a name, it is a instance for the class time at this point with two variables, one called hours and one called minutes. Time has been initialized by SELF. Okay, now there is terminology associated with this and in Python it is called the construction spirit and so whenever you have a class, you must have a construction. Whenever you have a class you must have a construction. Whenever you have a class you must have a construction. And that construction will look like this. Underscore INIT underscore SELF. Is that clear? Raise your hand if you have this. Make sure you are getting this so you cannot take a class and you can pass that to a variable may be called by time, and remember (INDISCERNIBLE) and initially you have two things, initialize zero, zero. And so you will do something like this, this time that we are talking about - and now you can use the name of that class and you can assign it to a variable. So, this variable will be referring to a class time that has two particular variables inside, one called hours and one called minutes. It is usually good to think about this as attributes. Attributes of the class. If that helps.
And so when you do this, the variable now, the value of this variable is a CLAS. Now, you can access now this variable, you can access the attributes, total hours and you can change it. You can set it to nine. And so what that will do is it will update this nine. You can take the second one, upgraded to 45 and so that would be upgraded to 45. You can take that CLAS and assign it to some other variable, time two, for example. So, here you have time one that we were talking about and here you can assign that to another variable. And then that variable will be initialize to whatever it is and now you can open it with time one and time two. As two completely different - as you have been doing with variables all the time, okay. So, you can change this. This is extremely detailed about this, there's nothing that you will not understand about this year. Now, there is a little bit of a difference now when you want to enhance this construction that we have before, before only we have DEF underscore INIT underscore SELF and now that is the normal way to initialize a CLAS. But, you can actually add parameters in that construction. So, you may have besides SELF, you may have all of the variable names here. And then you can open it with that as well. It is important to understand what this means. This one is associated with itself. This is a attribute of self. This is an attribute of self. And here you can access that attribute by saying SELF.hours and you can assign whatever it is to this and so this is parameter passing by assignment.
And now you can have a class or two parameters, nine and 45 and noticed there are one, two, three parameters and here we had time with two parameters and you have to specify SELF. And now you can operate with this. And now the initialization of the classes by doing this and so the interesting part of a CLAS is what do you do with the elements of that CLAS and so metrics are functions and so for example you can define some function that you would like to take a parameter from the class and you would like to check it this morning or not and so how did he do that? You take that class and you check the parameters and then you return - and it is true. Others parameter is less than 12. So, that means it is morning. There's nothing deep here, but it is important understand the fact that classes, the way they are defined you can assign them two variables. Now, there are two - and I want to

mention something that computer scientists like so you can understand where these things are coming from and there are two very particular data structures and remember you can think of CLAS as data structures, too. They are called stocks and queues. Have you heard of a stack? And have you heard of a qeue? These are two ways to maintain data and so when you have data structures you have to have two things. One is how the data is laid out in the case of a stock and in the case of a queue. And so that is the data laid out. And the most important thing about the data structure is how it operates so you have to follow a protocol.
And a protocol is how new stuff comes into the data structure. And so if there is a stack, the protocol is that this goes this way and that means that your stack is coming here. If it is a queue, this stack is coming here. So, this one goes this way and this one goes this way. Would you like to have a bank that uses this? (POOR AUDIO) of to write several pieces of code. This does it. What is the trick? What is the work being done? (POOR AUDIO) and so we know that we have a method called (NAME) and that pop we would like to (INDISCERNIBLE) and now you have the definition of a CLAS for queue that is routinely using computer science for many things. And you can look at the examples here. You are in a element, the element that your adding is here and you want to have a picture (INDISCERNIBLE) and the element comes and remember this is going this way. You want to take something out and you're going to take it from here. Okay. You can check the example. There is nothing here. The stock, you can now see how similar a stack is to a queue. But, in terms of calling this, it should be very to me I think you know a stack you will define it as a stock and we are going to initialize it. (INDISCERNIBLE) And we assign that something called several items. And so the data for this CLAS is going to be a list. This is what this is saying, this is nothing different than when we have a queue definition in the previous example. And now we are going to have something called push and pop. Well, look at how this looks. This does not look too different, does it? Self items is this list and we are appending something, a new item. The difference here is in this pop. Do you want to pop? Well, this is a list. And we know how to pop something. Now, the difference here and here is that over there we were popping the element at the zero position. Open the element as a zero position. (POOR AUDIO) and you can look at this. Okay, so for example - I think this is good enough.
>>SPEAKER: (AWAY FROM MIC) (POOR AUDIO) .
>>PROFESSOR: Section 11, section 10. The map is on the ball. Okay? So, can you talk to your group members? Yes, you can talk to your group members. But, at some point time is going to run out. So, you have about 13 questions and about 10 minutes we are going to ask you to commit to the answers, yes? You are confused? You are doing the midterm practice. Are you confused? Still? Okay, you have a piece of paper. Do you know what the piece of paper is for? To record your answers so that you can get points. You're going to get bonus points for this, yes? No, no, no, because you record the answers. . You are going to write your answers on that piece of paper. You are not recording the answers right now. Is that clear?
>>STUDENT: Limit -- (AWAY FROM MIC) (POOR AUDIO) .

>>PROFESSOR: (INDISCERNIBLE) Each question. Okay, the first question is 20. The next question is 21. The next question is 22, is that clear? (POOR AUDIO) you guys can go ahead. You will be recording your answers on that little piece of paper. (POOR AUDIO) what do you need? (POOR AUDIO) >>SPEAKER: (INDISCERNIBLE).
>>PROFESSOR: Your 29, right? -- You are 29, right? That is rubbish, one guy did not show? Okay, take your time.
>>STUDENT: How is the answer to this on here?
>>PROFESSOR: What is the question? (POOR AUDIO)
(INDISTINCT CHATTER)
>>PROFESSOR: (CHUCKLE) I see. (POOR AUDIO) .
>>PROFESSOR: Okay, I think it is time to report. We will report on whatever you have done. And so we need to go to streamline (SP?), that is the name. And you have to self-declared yourself. (POOR AUDIO) R