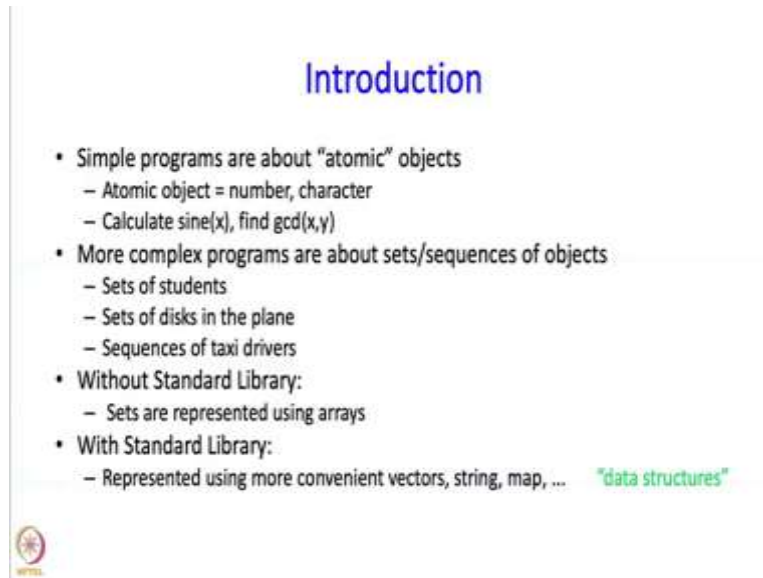


**An Introduction to Programming through C++**  
**Professor Abhiram G. Ranade**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology Bombay**  
**Lecture-24 Part-01**  
**Data structure based programming**  
**Introduction**


Hello! Welcome to the NPTEL course on an introduction to programming through C++. I am Abhiram Ranade and today's lecture is about data structure based programming.

(Refer Slide Time: 0:30)



**Introduction**

- Simple programs are about “atomic” objects
  - Atomic object = number, character
  - Calculate sine(x), find gcd(x,y)
- More complex programs are about sets/sequences of objects
  - Sets of students
  - Sets of disks in the plane
  - Sequences of taxi drivers
- Without Standard Library:
  - Sets are represented using arrays
- With Standard Library:
  - Represented using more convenient vectors, string, map, ... “data structures”



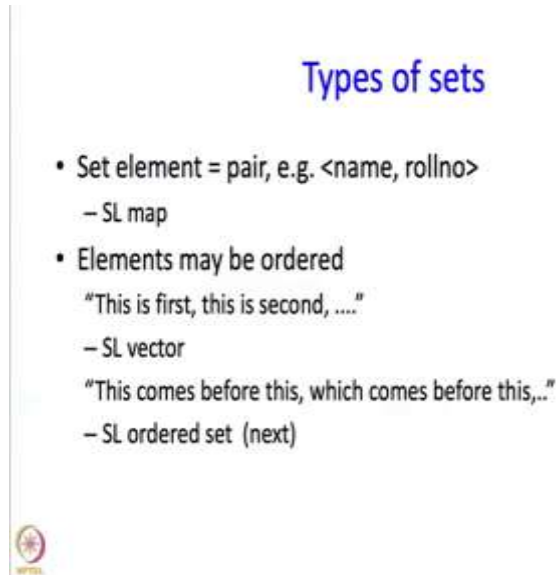
So, let me begin by saying that simple programs are about atomic objects. And by atomic objects I just mean numbers or characters. So, for example, we wrote programs to calculate the sign of a number or the gcd of two numbers, so these are what I have called simple programs. More Complex programs and perhaps most programs are about objects which are collections or sets or sequences of objects.

So, there will be sets of students we are dealing with in our program, or maybe some set of disc in the plane, some set of taxi drivers, but we are no longer just talking about one or two objects. We are talking about lots of objects and they even might have some relationships amongst them. Now, of course, these programs can be written without the standard library which we learnt about in the last lecture.

And typically we will represent sets using arrays. With the standard library, we have a much bigger choice, so we will represent sets using vectors, strings, maps and these are just more convenient ways of representing sets. And when we say data structure based programming,

what I mean over here is programming which involves things like these vectors, maps, and so on.

(Refer Slide Time: 2:00)



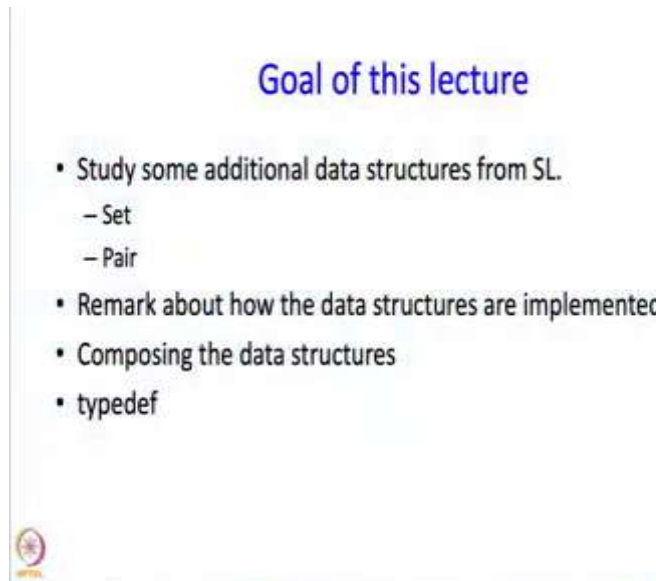
The slide is titled "Types of sets" in blue text. It contains two main bullet points. The first bullet point is "Set element = pair, e.g. <name, rollno>" with a sub-point "– SL map". The second bullet point is "Elements may be ordered" with two sub-points: "“This is first, this is second, ....”" and "– SL vector", followed by "“This comes before this, which comes before this...”" and "– SL ordered set (next)". There is a small logo in the bottom left corner of the slide.

- Set element = pair, e.g. <name, rollno>
  - SL map
- Elements may be ordered
  - “This is first, this is second, ....”
    - SL vector
  - “This comes before this, which comes before this...”
    - SL ordered set (next)

Ok, so when we talk about sets, there are different variations possible. So, for example; the elements of the set might be a pair. So, for example; you might be storing pairs of the form, name and roll number of that student. So, this is very nicely represented using a map data structure or a map class in the standard library. SL in this lecture is going to mean standard library.

Another possibility is that not only do you have a set of objects or set of entities but they may be ordered. So you may think to yourself that this is the first entity or this is the first object, this is the second object, this is the third object and so on. In such case, the standard library vector class would be extremely useful. On the other hand, in the relationship is not first, second, third but just that this comes before this. This in turn, comes before this other one and so on. So, there is just an ordering relationship in which case the standard library class called ordered set actually it is just called set but by without any prefix to it, it means implicitly ordered set. That class will be very useful or that data structure will be very useful and we are going to see this very soon.

(Refer Slide Time: 3:39)



The slide has a light blue background. At the top center, the title 'Goal of this lecture' is written in a bold, dark blue font. Below the title, there is a bulleted list of five items. The first item is 'Study some additional data structures from SL.', followed by two sub-items: '- Set' and '- Pair'. The second main item is 'Remark about how the data structures are implemented'. The third is 'Composing the data structures'. The fourth is 'typedef'. In the bottom left corner, there is a small circular logo with a stylized 'S' and the word 'SPEECH' below it.

### Goal of this lecture

- Study some additional data structures from SL.
  - Set
  - Pair
- Remark about how the data structures are implemented
- Composing the data structures
- typedef

So, in this lecture we are going to study some additional data structures or additional classes from the standard library and these are going to be the set and something called a pair. Then, we are going to see how the data structures are implemented. This is going to be just about a remark because going into details would be quite complicated.

Then we will see how these data structures can be composed, because the real power is obtained when you can compose these data structures. Then we will see a simple primitive called typedef, but all of these we will see in the subsequent segments, so we will take a quick break here.