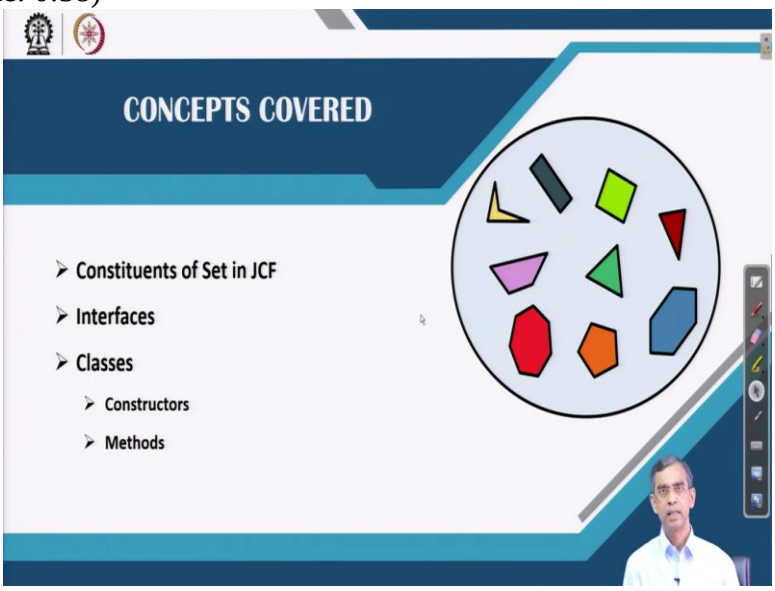
Data Structures and Algorithms using Java
Professor Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture 08
Topic: Set of JCF

Java C, a collection from the different viewpoints. In the last video lectures we have studied few such views like a collection is viewed as an array list or it is as a link list or Q or dQ like this one.

(Refer Slide Time: 0:58)

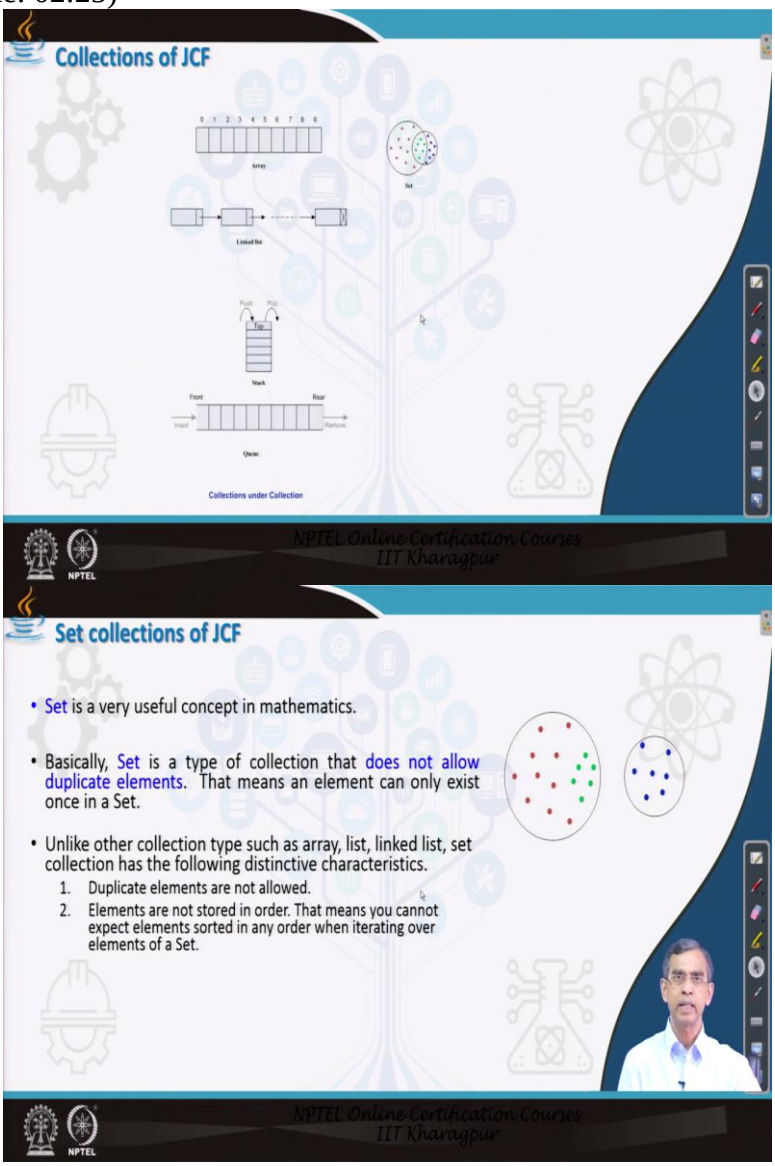


Now, today there is another view. This view is called the set view of a collection. Set basically is a very common word. Set, you know, it is basically, is a group, is a collection again, whatever the name you can say. But the difference from other collection to the set is

that it basically does not allow to include any duplicate elements in it. So, whatever the elements are there, all the elements are to be unique, no duplicate element should be there.

This means that if a set is basically a set of names, then there should not be two names having the same. Now, like the collection concept like other structures, the set also defines a number of interfaces and classes. So, in this class, we will learn about whatever the interfaces are there in the set collection and the classes. Classes as the class contains constructors as well as the methods. We shall discuss what are the constructors and methods are there in some classes those are belongs to the collection called set.

(Refer Slide Time: 02:25)

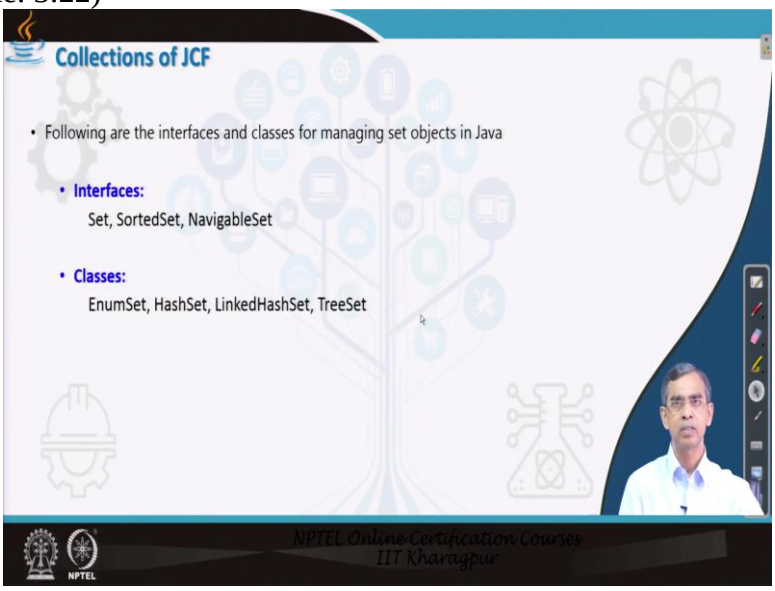


Now, set is another kind of data structure as I have already told and I have already mentioned that set can be represented either using indexed manner or in the sequential manner. But in addition to this, Java supports to maintain the set either using index means look like an array

or is a sequential, look like a list. Also, it can allow you to include in a very peculiar manner, it is unique or novel manner actually.

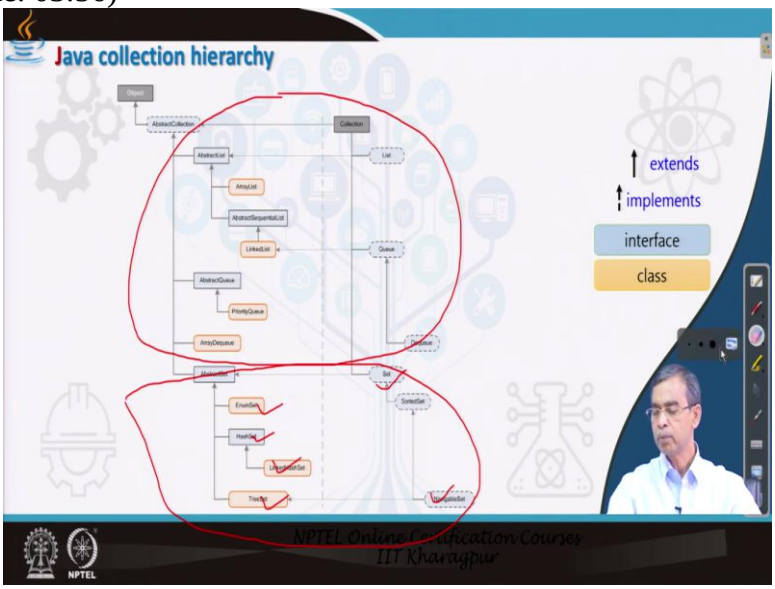
They are called tree form. Concept of tree is very important for, from the data structure point of view. So, tree concept will be discussed in this class later on but tree is a very concept. So, the set also can be represented using this concept 'tree'.

(Refer Slide Time: 3:22)



Now, so far the interfaces are concerned in set collection, there are three different interfaces namely, set, sorted set and navigable set. There are few classes which basically implements all these sets interface namely, EnumSet, HashSet, LinkedHashSet and TreeSet. So, we will try to have a quick view of all the sets those are there.

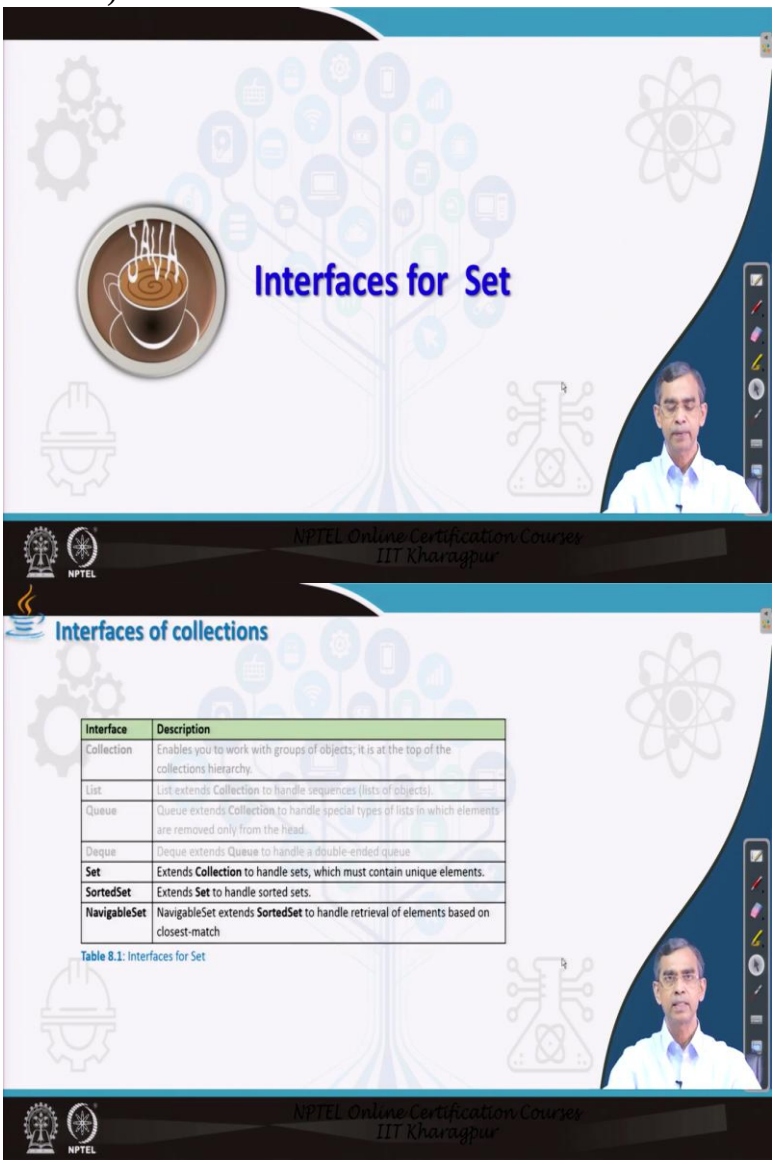
(Refer Slide Time: 03:56)



Now, again you see, we have discussed in the last video these parts, whatever the interface and (())(4:04) up to this one. The next part that we are going to discuss in this class is basically these are the things are there. So, these are the interface sets, sorted set and navigable set. And the classes those are there is EnumSet, HashSet, LinkedHashSet and TreeSet out of which HashSet is basically abstract class.

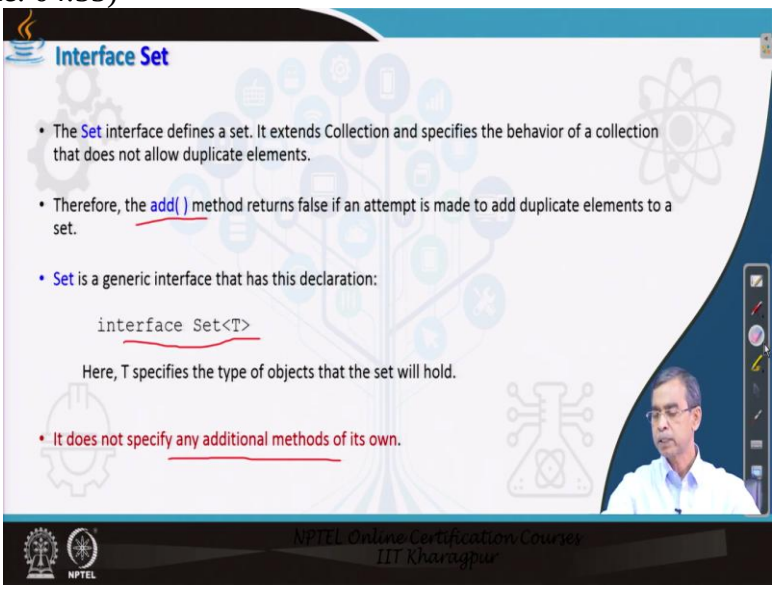
These are the basically useful class for which you can create an object and then for that object you can invoke all the methods which are defined in those classes.

(Refer Slide Time: 04:47)



Now, let us see what are the interfaces are there in each of the interfaces that we have mentioned namely, set, sorted set and navigable sets.

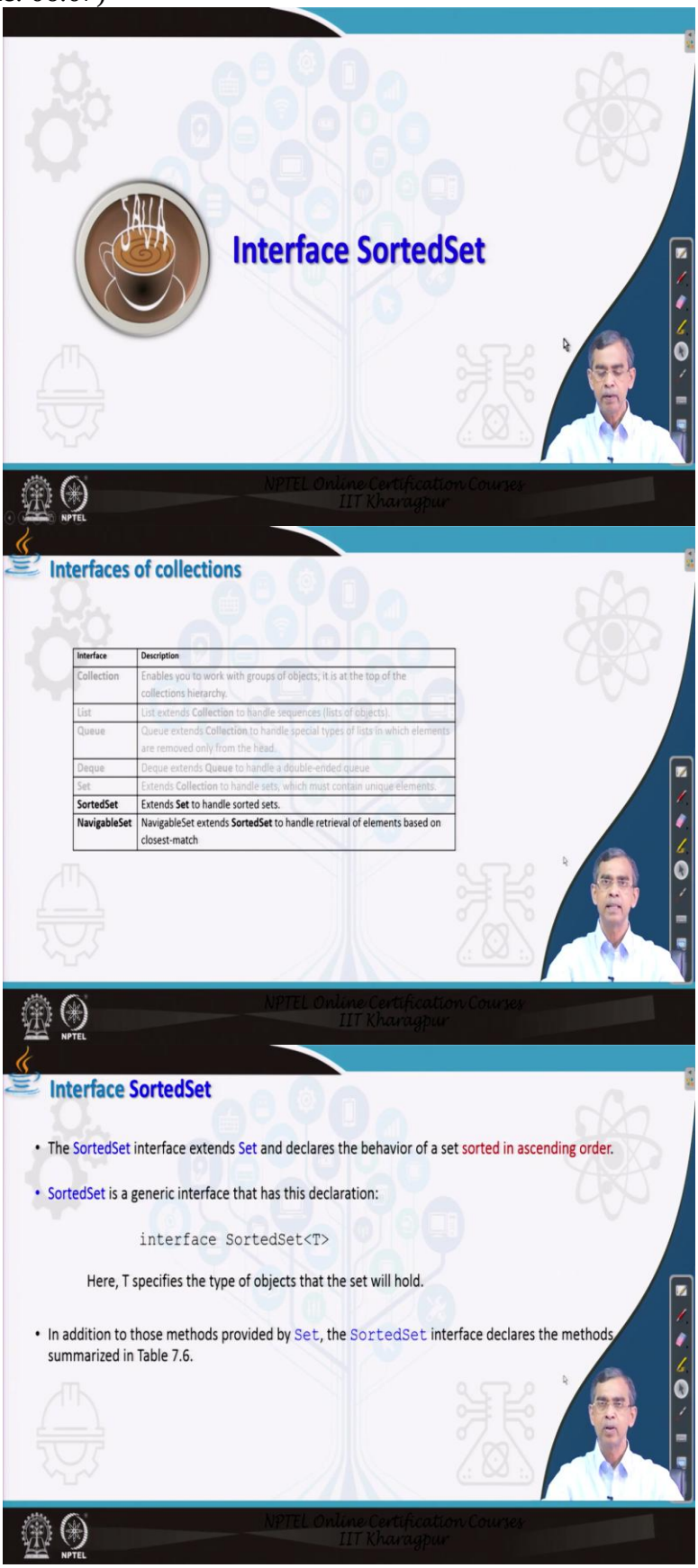
(Refer Slide Time: 04:55)



Now, again like other collections that we have studied so far, set is also a generic form. That means it allow to include elements of any type. That is why a set is basically interface. On generic type it is there. So, only one type of collection it can store. So, only one template is specified here and the interface sets has the, say in add methods which is basically same as the add method that is there in collection because set is also, is an extension or is a child class of the collection.

And another thing is that in the interface sets there is no any additional methods of its own other than all those methods by virtue of inheritance those are inherited from the collection. So, those all the methods those are there in the collection you can call for the set collection actually. So, that is why and there is no specific method defined in the set as an interface.

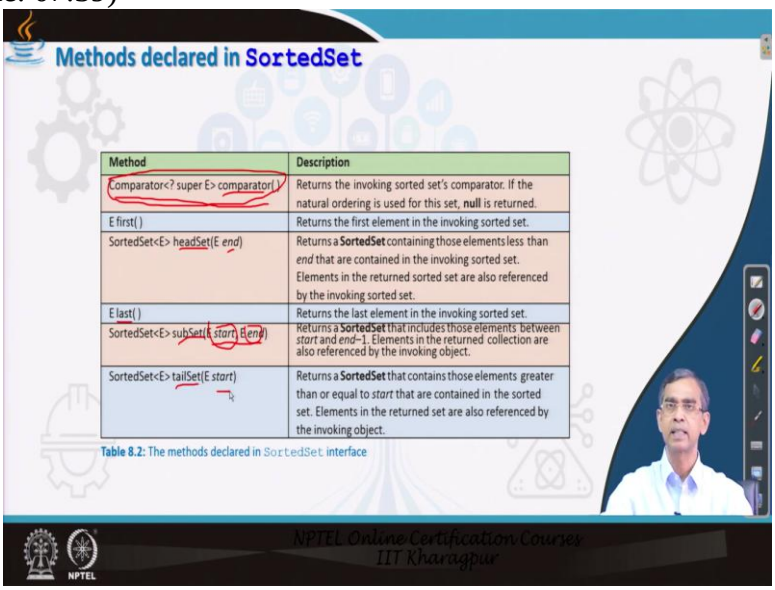
(Refer Slide Time: 06:07)



Now, let us see the interface sorted set. Set is a collection where all the elements are not necessary that to be arranged in an order. On the other hand, if it is a sorted set then the elements are to be arranged in an order. So, that is why sorted set. So, a set and sorted set, they are basically same. Only difference is that in the later one the elements are sorted in an order. Usually, the sorted set is more meaningful for the data like integer, float or string.

But for other user defined data type like book or student they are not meaningful unless you define to be a meaningful from the sorted point of view. Now, like other collection sorted set is also a generic collection. That means it can include any type actually but they are basically limited to numeric if you have by default. But you can customize them writing some methods of your own to make it applicable to other defined data type.

(Refer Slide Time: 07:39)

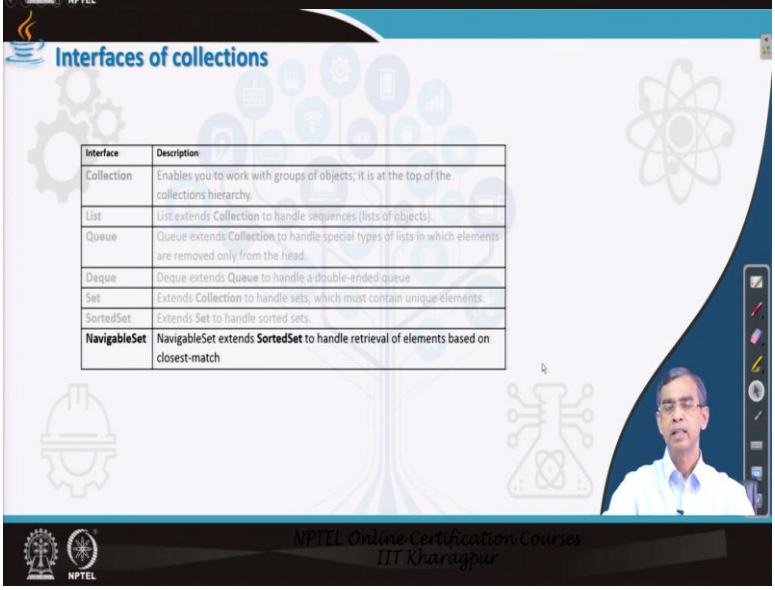


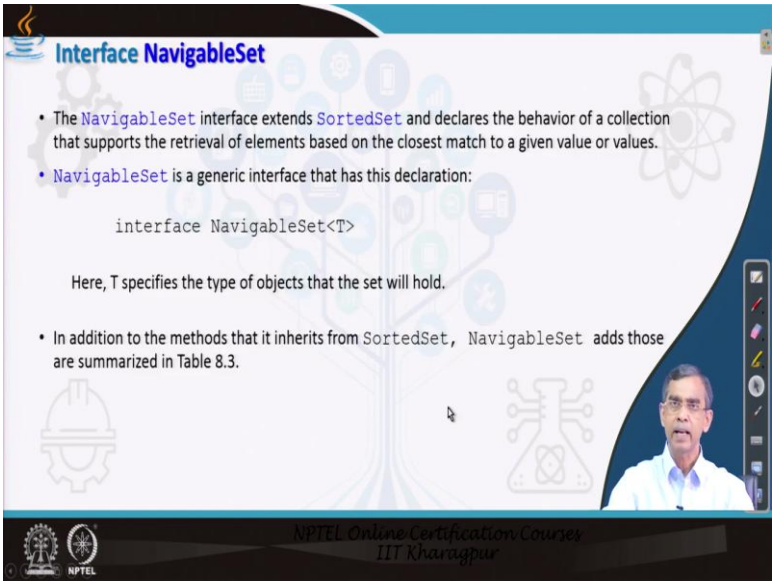
Now, let us see what are the methods are there in the sorted set. It is basically similar to the others. Now, here is a comparator method, comparator on method because if you want to make say student as objects to be stored in a sorted set, so you can define or you can modify the comparator method. Comparator method is defined in the collection interface actually. But we can modify comparator method to be applied to the sorted set also.

That is why this comparator. Comparator is basically that how you can compare one object to another from the sorted point of view. Now, head set is basically written the elements which contained in this current elements starting from the end element. So, this is the one method. The last is basically returns what is the last element that is stored in this collection. Subset as the name implies it basically returns the subset of the elements.

The subset is basically has a range that from the element which having the starting at start and ending at this one. So, it basically returns all the elements between start and end both inclusive. And it is basically tail set just like subset only but tail set means a particular portion of the set starting from a particular elements in the set. So, these are the methods are there to access or to manipulate your collection in the different way. So, this is about sorted set.

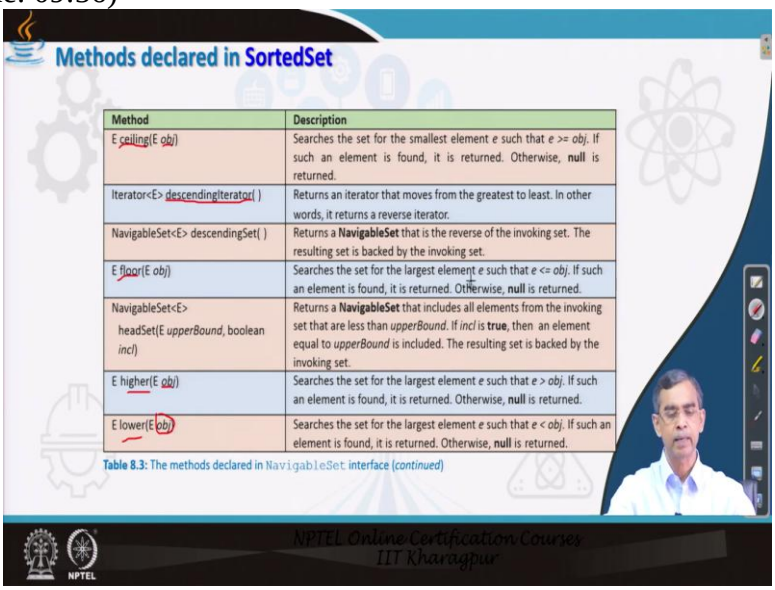
(Refer Slide Time: 09:26)



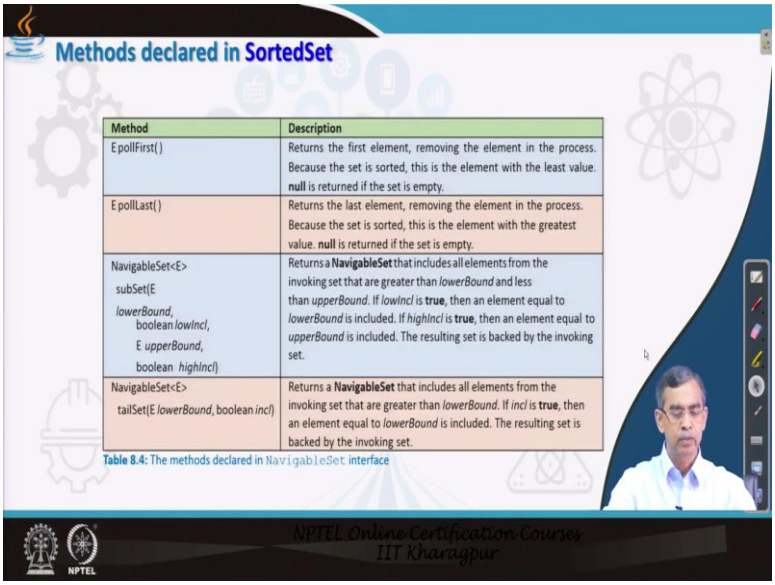


Navigable set is in fact another advancement of the sorted set but it is not necessary to be sorted there. But navigable is from the searching point of view for the faster searching to a collection if you want to have, then you can think for storing this as a navigable set. Now, navigable set includes any type of data like other collection and it has many methods defined in it.

(Refer Slide Time: 09:56)



Method	Description
E ceiling(E *obj*)	Searches the set for the smallest element *e* such that *e* >= *obj*. If such an element is found, it is returned. Otherwise, **null** is returned.
Iterator<E> descendingIterator( )	Returns an iterator that moves from the greatest to least. In other words, it returns a reverse iterator.
NavigableSet<E> descendingSet( )	Returns a **NavigableSet** that is the reverse of the invoking set. The resulting set is backed by the invoking set.
E floor(E *obj*)	Searches the set for the largest element *e* such that *e* <= *obj*. If such an element is found, it is returned. Otherwise, **null** is returned.
NavigableSet<E> headSet(E *upperBound*, boolean *incl*)	Returns a **NavigableSet** that includes all elements from the invoking set that are less than *upperBound*. If *incl* is **true**, then an element equal to *upperBound* is included. The resulting set is backed by the invoking set.
E higher(E *obj*)	Searches the set for the largest element *e* such that *e* > *obj*. If such an element is found, it is returned. Otherwise, **null** is returned.
E lower(E *obj*)	Searches the set for the largest element *e* such that *e* < *obj*. If such an element is found, it is returned. Otherwise, **null** is returned.

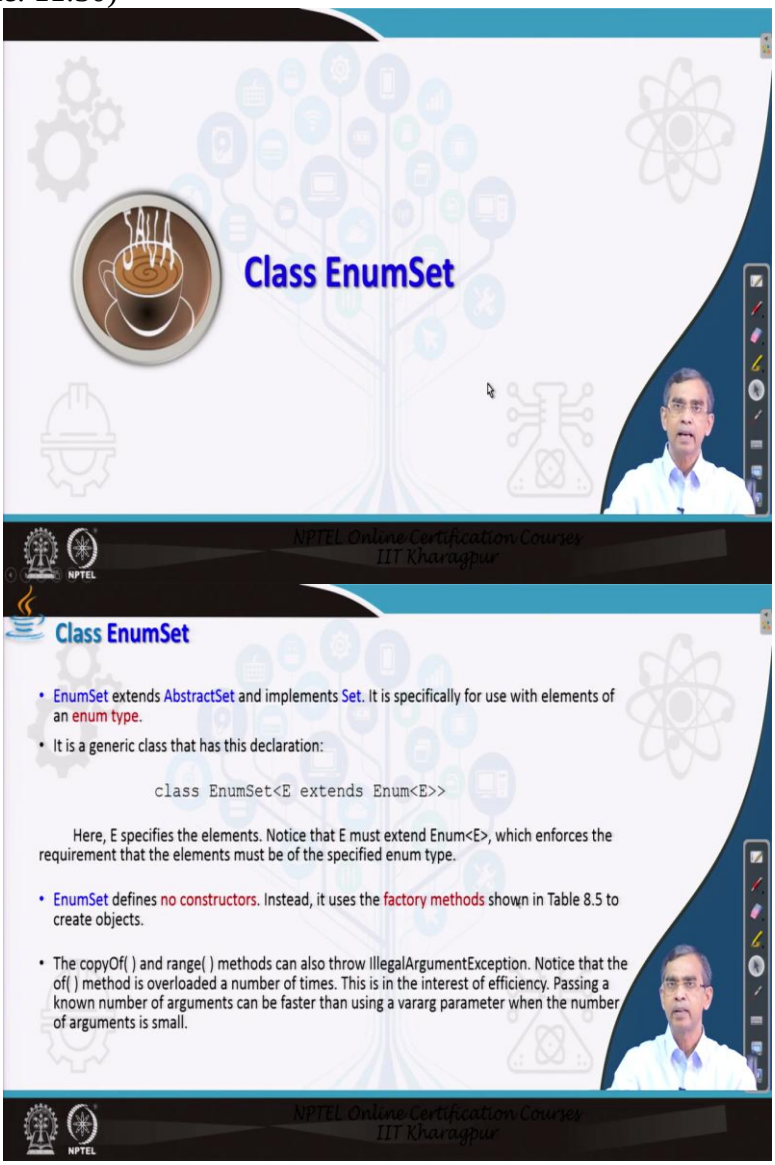


Few methods just for discussion like a ceiling. Ceiling is basically is the method, is basically starting from a certain elements it will basically return all the elements. Ceiling means from a particular portion of, particular point of the element it will give all the rest of the element like.

And descending iterator is basically traversal of the collection in a descending order. And then floor is basically like ceiling there is another, starting from a certain it is basically ceiling if it is top to bottom, then it is from bottom to top traversal actually or returning like.

And then higher is also similar to ceiling and floor. It basically return some elements which is larger than a current element which is passed as an input. Lower is also similar to floor is basically which element has the largest element which is less than the object actually it is there, passed as an argument. So, this way from the collection which is stored as a sorted set we can access the different elements from there. There are few more methods that we have discussed here.

(Refer Slide Time: 11:30)

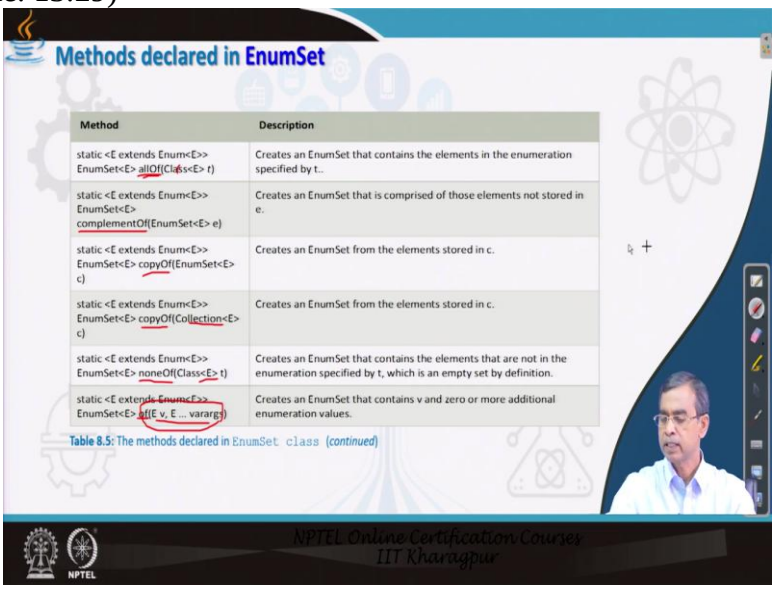


Now, there is another class which belongs to the set category, it is called the EnumSet. Enum Set concept is basically it is a collection of some enumerated data type. Enumerated data type means some data types which does not belongs to any type of data but it is called enumerated. For example, seven days of a week can be enumerated as Sunday, 0; Monday, 1, to Saturday, 7, so it is called enumerated. Similarly, say 12 different months or eight different colors, all these can be enumerated form or size of a T-Shirt can be enumerated form.

Different concept it is there. So, if you do not know enumerated type of data or it is very popular in C and C plus called Enum type, so Enum data type. That means it basically define a certain data type, it is called the enumerated. So, Java collection also facilitates how this kind of enumerated type elements also can be managed. For these things there is a class called EnumSet.

EnumSet class does not have any constructor of its own but only few static methods to create enumerated set. And there are few methods, those are very popular to copy some elements from this enumerated set namely, copyOf, or range method like this one. All this enumerated set and then their application will be discussed when we will discuss about application of this Java collection framework to a real life data structure.

(Refer Slide Time: 13:29)

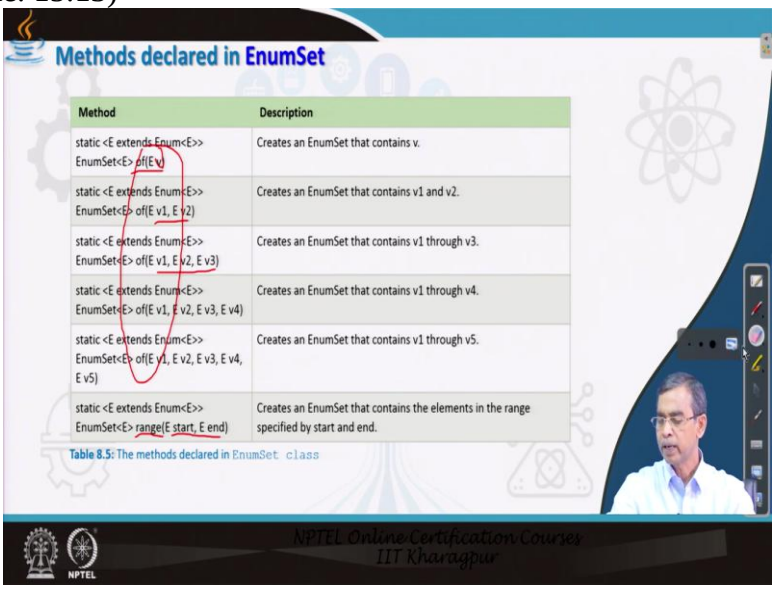


And it has certain factor E method it is called, those methods by which you can create enumerated type of data and here is basically is that this is all of, that means is a method by which a set can be created. I say, I told that it does not have any constructor. So, there is no constructor but if you want to create a set collection, then allOf method it basically pass different objects and then it can add it into this and then this way the enumerated set can be.

Like allOf it is compliment Of, that means it is basically if a is a set, then a compliment. That means those are not there in a but they are in some other universal set it is like. And copyOf is basically copy a particular collection into another EnumSet like. This is another way, copyOf from the existing collection which is passed there. noneOf means it basically give the collection into this collection which is not there rather than this one.

It is just like complement for some. And it is also, there is another called Of method which basically is a, used as a variable things, means it says basically from a certain enumerated set v 1, v 2, v 3, within a certain range it can have copy of it. So, these are the different methods by which you can create, you can maintain the Enum, enumerated type of sets.

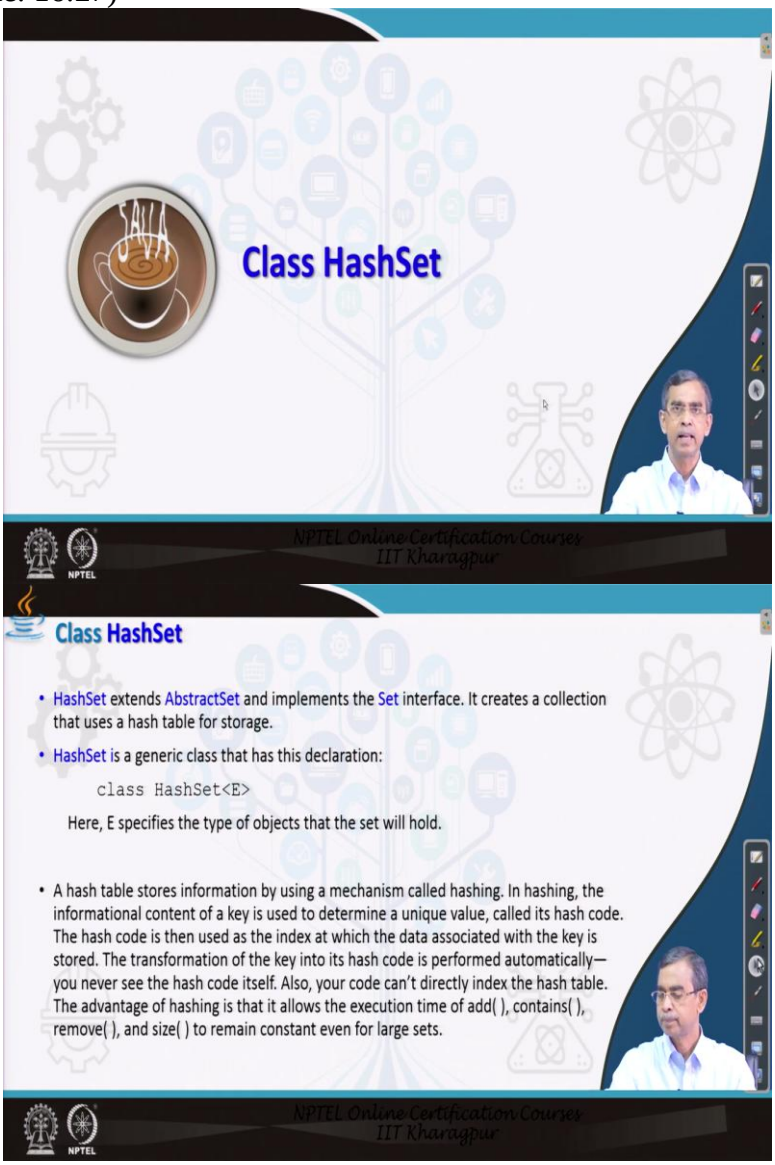
(Refer Slide Time: 15:13)



It has few methods they are also declared. Those methods in addition to the method that we have discussed, for example, of E v, so if particular element v is there, it basically check that element v is there or not. Now, v 1, v 2 for the two EnumSet elements is there. v 1, v 2, v 3, it check that whether v 1, v 2, v 3 are there or not. So, it is, there is a different version of Of method as we see whether a particular Enum elements is there, two element, three element, four element and maximum five elements it allows.

It also have one method called the range method. It basically check or it basically that creates EnumSet which contains all the elements in the range specified by start to end. So, it is basically creating another set, EnumSet using this method. So, these are the few methods that those are, okay, they are in order to maintain the Enum Sets. But learning of all those methods will be more clearer whenever we can include some examples. Definitely, all the examples will be covered while discussing details about this EnumSet class.

(Refer Slide Time: 16:27)



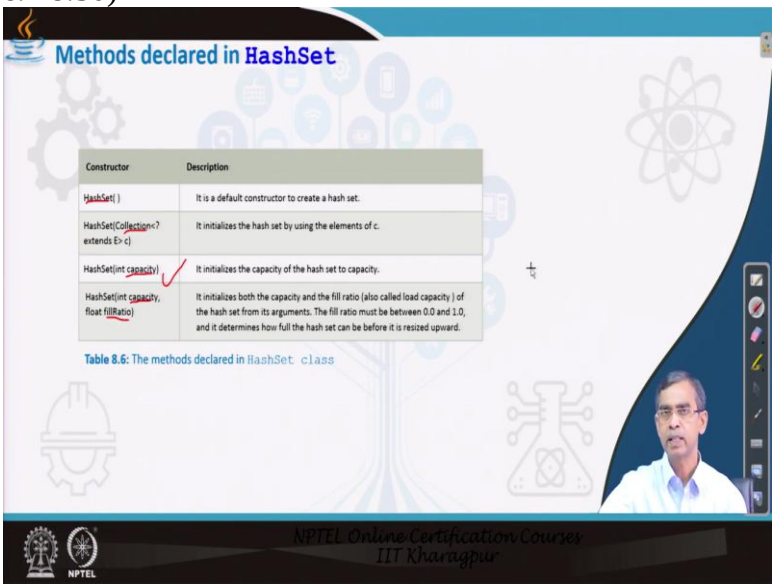
Now, we discuss about another very interesting collection, it is called the HashSet. Now, concept of hash is a very important one concept and it is useful in many application and particularly called the hashing application. Hashing is the concept which basically have been introduced to retrieve an element from a collection in a most fastest way. So, for faster retrieval of an element. Now, how it is possible that concept is followed hash.

Hash is basically for every element it creates one hash code. Now, hash code is something by which a particular element can be proved in a collection whether it is present there or not. So, whenever you want to search for an element you just create its hash code and then list the table of all the hash values and then you can check whether element present there and where it is present. So, this is the concept, now regarding the hash code concept we will discuss these things in details when we will study the table data structure or map data structure again.

Now, whatever the thing it is there for an element to be accessed, so there is a hash value to be considered. Now, hash value can be created by a method hash code which is basically, is a unique for every unique element actually. It is not true that for two different element the same hash value. It is never. So, this is the way for every element a hash value be created. For example, if your name is Saurabh and if you want to apply the hash code to this, for Saurabh it will return a unique value, hash value.

So, this concept that is called the hashing concept and if you store a collection in addition to the utilization of the concept hash, then that collection is called the hash collection. Now, it is more precisely called hash set collection.

(Refer Slide Time: 18:50)



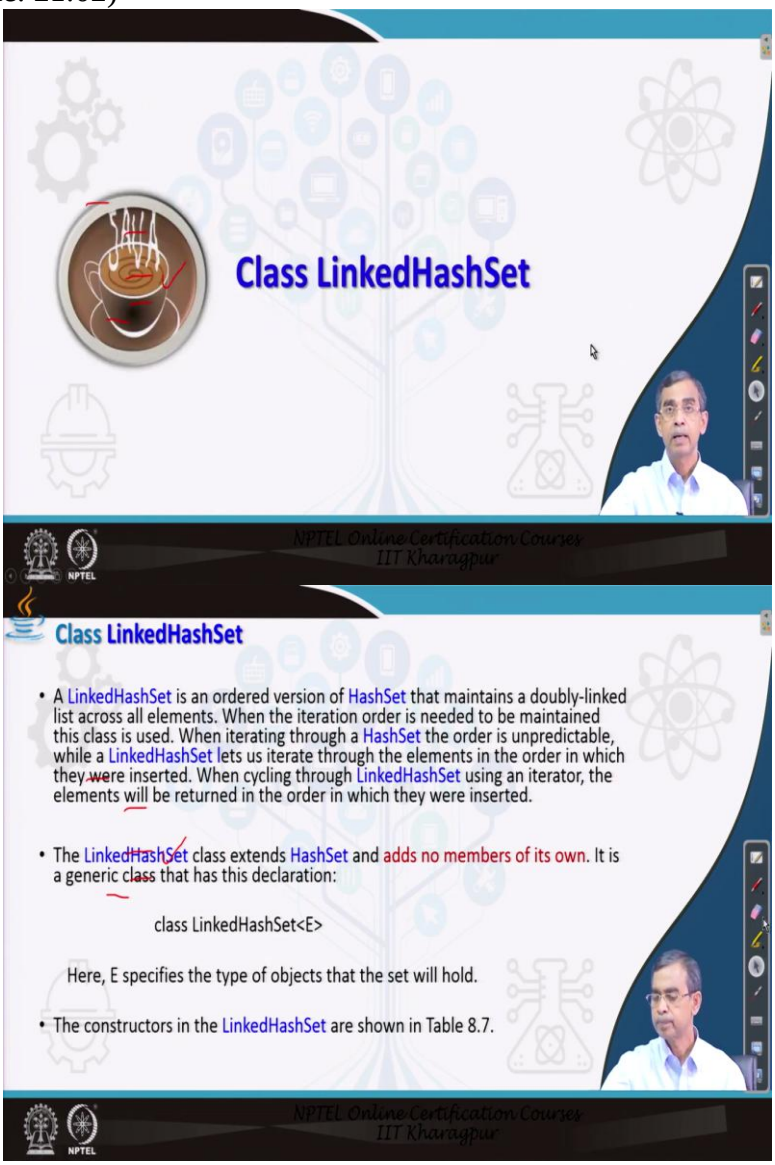
Now, hash set collection if you want to create, so for creating hash set collection there are few methods, constructors are there like hash set constructor it is basically is a default constructor. Hash set for a certain collection, for a set of numbers if you want to store them as a hash set, then this constructor can be created. And you can create a hash set giving an integer value as a capacity. Suppose we want to create a hash set collection with 100 size or capacity, then this constructor can be there.

Another way of creating a hash set mentioning the size as well as the fill ratio. What is the concept? Is that if you have decided that one hash set you initially create with size 20, later on you keep on adding, adding. Then what will happen after 20 is over? So, it basically dynamically grow the hash set collection actually. That means automatically it will increase the size, you do not have to tell.

So, if you initially fix their size 20 but later on if you want to store 30, no issue for there. Now, whenever you are inserting or adding elements into the hash set, it needs to be grow automatically. Now, that grow you can control by mentioning fill ratio. That means how much. So, if you say that 75 percent whenever the element is filled according to your mentioned capacity, then only it will be increased by another jump, maybe another size 10 or 20. Those things also you can mention.

So, this way you can mention the fill ratio it is there. So, fill ratio value can be anything above 0 to less than 1.0. So, in between 0 and 1.0 actually the fill ratio. As I said, fill ratio 0.75 means that 75 percent when it is full, then you increase the capacity of this hash set. So, this concept it is followed to automatically grow the hash set collection for your storage.

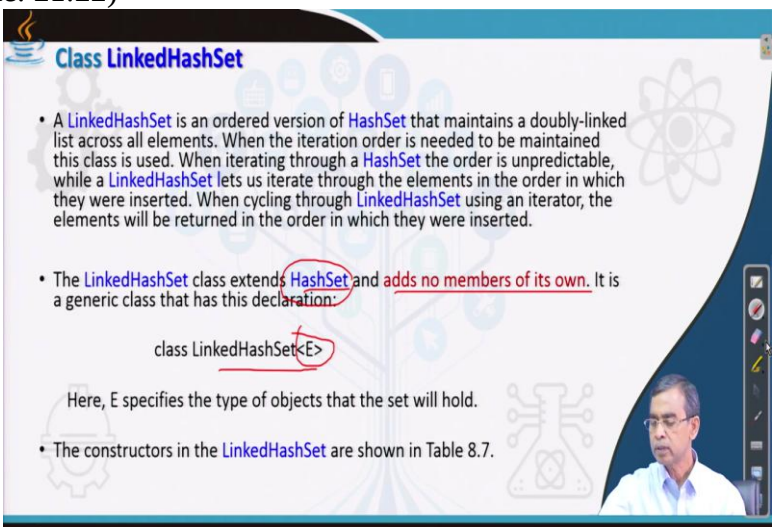
(Refer Slide Time: 21:02)

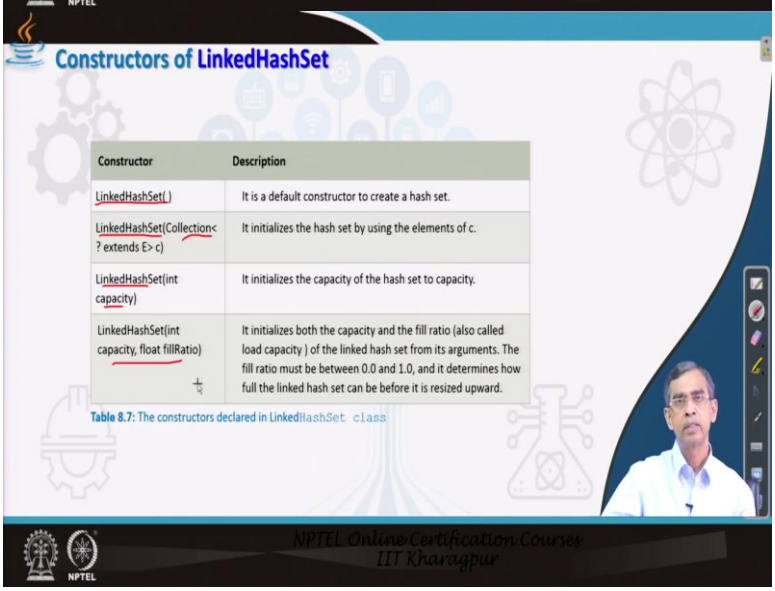


Now, linked hash set, it is the one way by which the sequential representation of the hash set actually. So, that means it is stored in a sequential just like a linked list form. Now, so, now linked list hash set is basically one form of hash sets and it is basically extends, extend the hash set. Hash set is one class that we have discussed. This is another form and this linked hash set does not have any member of its own.

Whatever the methods those are there in hash sets are basically method of linked hash set basically. Like hash sets linked hash set also store generic type of data. That means the template is there. That means it can store any type of objects that you want to store. Now, in addition to the methods those are there in the linked hash set, that means the same method of hash set it also include some constructors of its own.

(Refer Slide Time: 21:22)

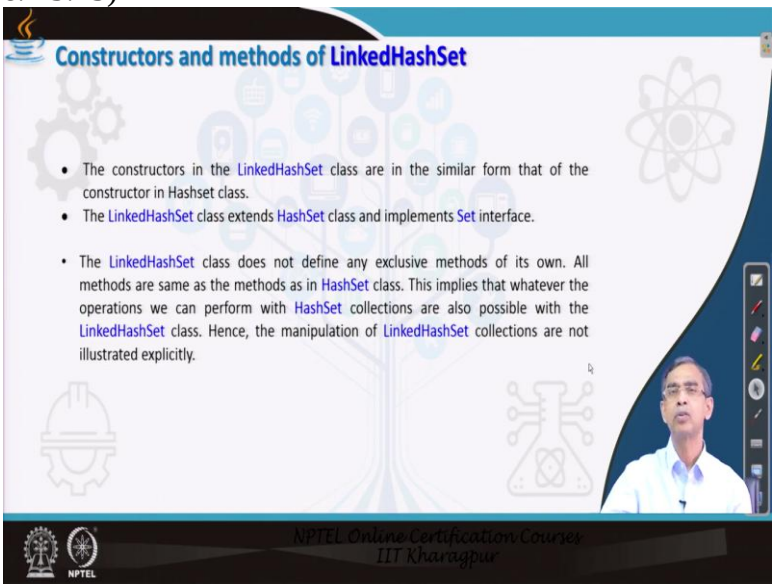




Now, the constructors which are there in the linked hash sets are tabulated here, listed in this table. Linked hash, this is the default constructors and this basically is same as hash set constructors. Basically if we want to create a linked hash sets with input as a collection existing and then this is basically same as the hash set, it is basically giving a capacity as the initial size and then it is also telling that automatically if linked hash set to be grow, so what should be the fill ratio that it can grow?

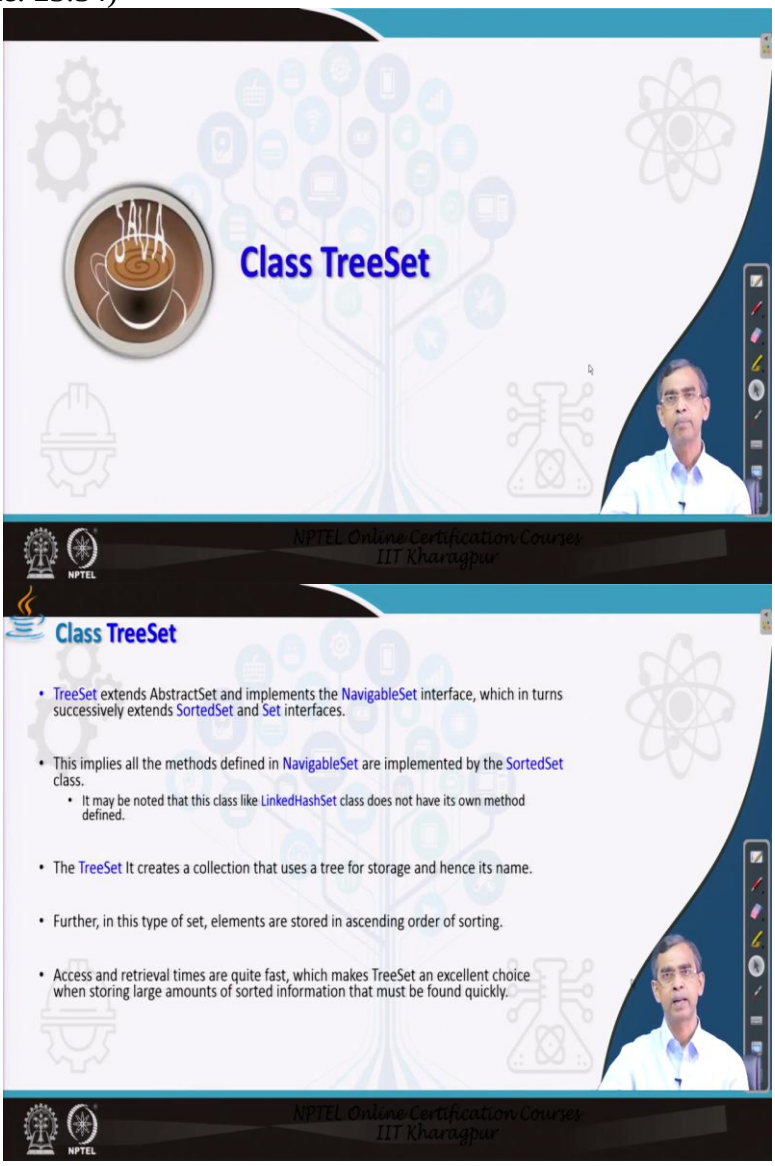
So, these are the different methods, different constructors are there and as I already mentioned, all the methods which are there in the hash sets are also there in the linked hash sets.

(Refer Slide Time: 23:15)



Now, so constructors and, hash set and linked set more, is very similar to each other actually. Only the thing is that internally they are maintained in a different way where that issues are need not to be bothered by the programmer.

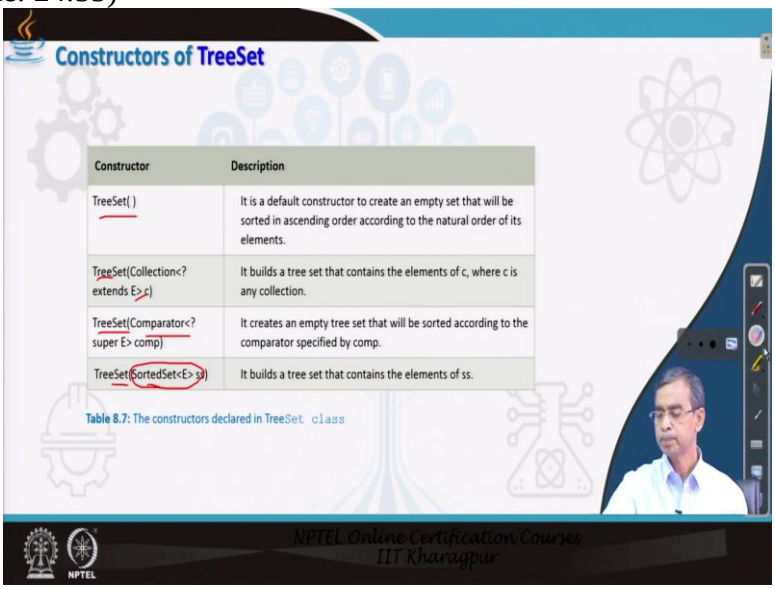
(Refer Slide Time: 23:34)



Now, there is another collection. This is called the tree set. It is a class, that means all the methods in this class are already defined and tree set is basically extension of set interface. It basically implements interface sets. But here it basically stored in the form of a tree. So, tree set extends one abstract called Abstract Set and implements the navigable set that we have discussed earlier. And it basically is a different internal representation of the collection in the form a tree structure.

The tree structure concept is a different than the array structure or linked structure actually. Now, the alike set a tree usually allowed to insert or delete an element keeping a sorting order actually. And further, tree set is another data structure which allow a programmer to access the elements from the collection in a fastest manner like other hash set like. So, this is the one collection is there in the tree set.

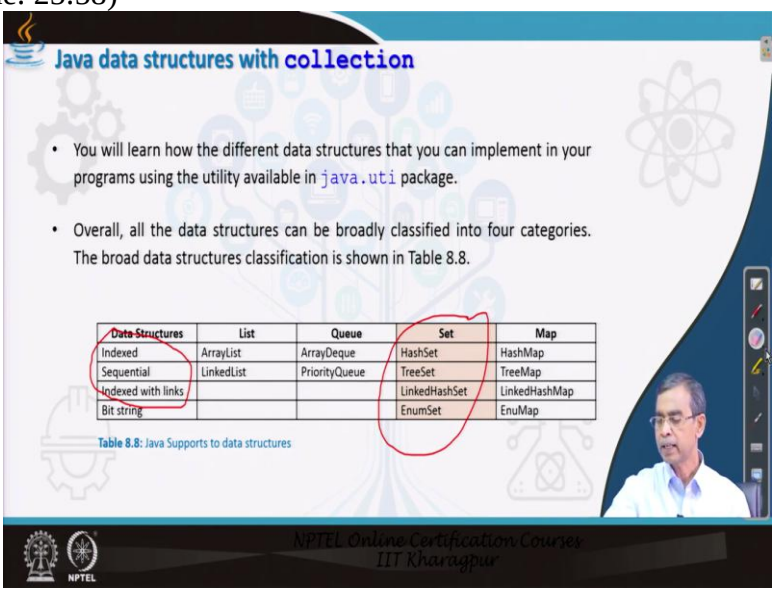
(Refer Slide Time: 24:55)



And the constructor those are there if you want to create a collection following the tree set form, these are the constructor. This is the default constructor and this is the constructor tree set, means if you want to create a tree set collection using an existing collection of any type, and then this is a tree set because it is needs to be stored in a sorted fashion. So, one comparator method needs to be defined and to be mentioned as an input that which comparator by which you want to store the element in the tree set collection.

And tree set also can be used with the existing sorted set. Sorted set is another concept where the ordering is mentioned. So, tree set is the one way that the sorted set can be given as an input to create a tree set collection. So, these are the different methods. Constructors those are there in the tree set and few classes.

(Refer Slide Time: 25:58)



This, okay, so these are the different methods, different interfaces and collection so far the set collection is concerned. Now, in summary what we can say that a set can be maintained using either hash set or linked hash set or tree set. And they also, there also a possibility of maintaining a collection in the form of a set but it is a enumerated type of elements. So, these are the concept that it is there.

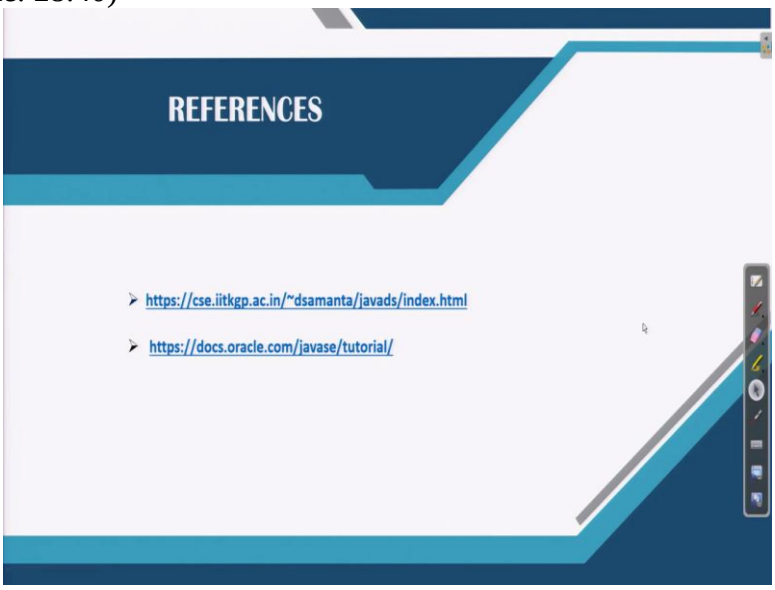
Now, out of which hash set is basically is a indexed based strategy. Tree set is a sequential strategy, linked hash sets is indexed with link strategy. And the enumerated set is basically the data structure that it consider is called the bit string. Now, all these data structure, indexed, sequential, or indexed with links and everything that will be discussed while we will discuss the specific concept of data structure, their theory.

So, their theory is an important aspects. Now, here we have discussed about the different collection, how they can be maintained and whatever the operations or functions in order to maintain them, that means functionalities. Functionality regarding how a collection can be created, how an element can be added, how an element can be removed, how status can be checked, how a collection can be traversed.

That mean all the elements can be visited one by one and so many things are there. So, these are the essential operations but behind these operation it will be very nice if we have good understanding about theory or practical or actual logic of implementing this application. So, once you know then it will be really give and adds to a very skilled programmer actually. So, that is why we should study little bit theory and then the application of those theory, that means detailed coding concept.

And using coding the conventional code programming as well as coding using or taking the advantage of all the Java collection framework those are declared there in Java dot uti package.

(Refer Slide Time: 28:40)



Now, regarding the today's discussion you can have the study materials either it from the Oracle tutorial websites or from the webpage that I have mentioned for you in the first link that is given here. So, I advise you to check the link, get the materials there, study according to your own space and learn more. If you have any further questions or you need some clarification based on any discussion, you can post those question in the discussion forum.

And one more important thing that I want to mention, week-wise assignments those are particularly related to every week's discussion you try to get the answer of your own and then verify your answer that you are confident about answer because attempting or answering all those questions really gives more understanding. Okay, fine. Thank you very much.