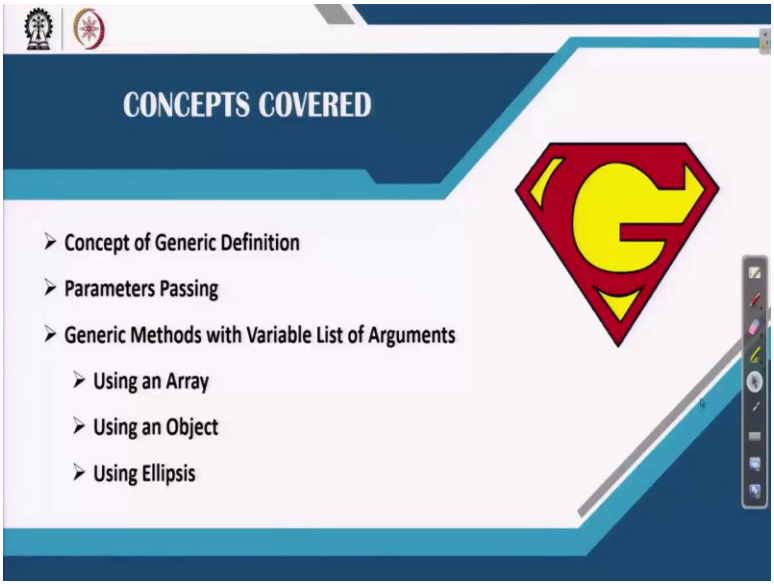
Data Structures and Algorithms Using Java
Professor Debasis Samanta
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
Lecture 2
Generic Methods

Okay, so we are going to discuss about generic programming. The topic is, in fact, vast topic and we have planned to cover this topic in conjugative 4 lecture periods, so this is the first of the series. Here we will cover generic methods, so generic methods of generic programming.

(Refer Slide Time: 0:58)



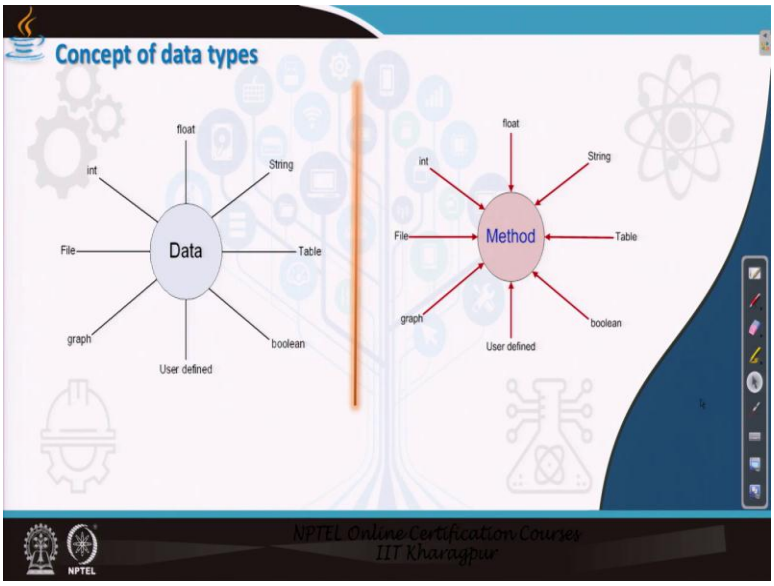
So plan of these lectures is like, first we should discuss about what is the concept of generic definition and then as we said in programming we have to write methods. Methods are basically operations or procedure to manipulate data and whenever methods are to be invoked we have to pass parameters to those method, so how the parameter can be passed to the methods, which will be defined in a generic way, so it is a parameter passing.

Now there is another concept in generic programming, so far generic method implementation is concerned, so with variable list of arguments passing, so that means one method can be called not a fixed number of arguments, any number of arguments, sometimes zero, sometimes 10, sometimes 2, sometimes any value, so this is called a variable list of arguments, how we can

define methods with variable list of arguments. There are 3 ways using arrays, using ellipsis, and using objects.

So three methods will be discussed in this course, in this lecture, we are going to learn about it. Now, what I want to say before going to actual discussion is that generic methods is a concept, which if we follow, then give the advantage of a programmer as a super programmer or the program that you can build is a super program you can build. So generic method gives a huge strength to your programming concept, and then you can build a program, which can deal any type of data irrespective of its type, that is the basic concept.

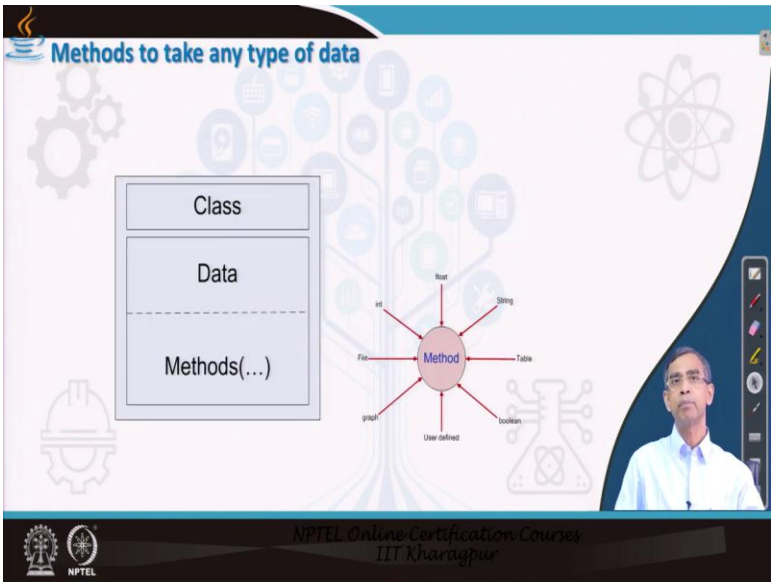
(Refer Slide Time: 2:52)



Now, let us come to the concept in details and the first thing that I want to say that at the moment data those are available are in different forms, like data can be in the form of integer, it can be form in the floating point values, it can be form in the form of a character or string and there are many other complex type of data, the table, graph or whatever it is there. And what exactly the need is that a method should be in such a way that whatever be the data you do not have to bother about.

In other words, so far the concept or the traditional programming is that, if you write a program or a method to handle a particular type of data, then it is limited to that (())(3:43) data, but how it will be, if we want to write a method, so that whatever be the type of data we can pass it to it will work for you. So if we do these things, then we can say that the method is a generic that means not that particular type, it is basically in general to all type.

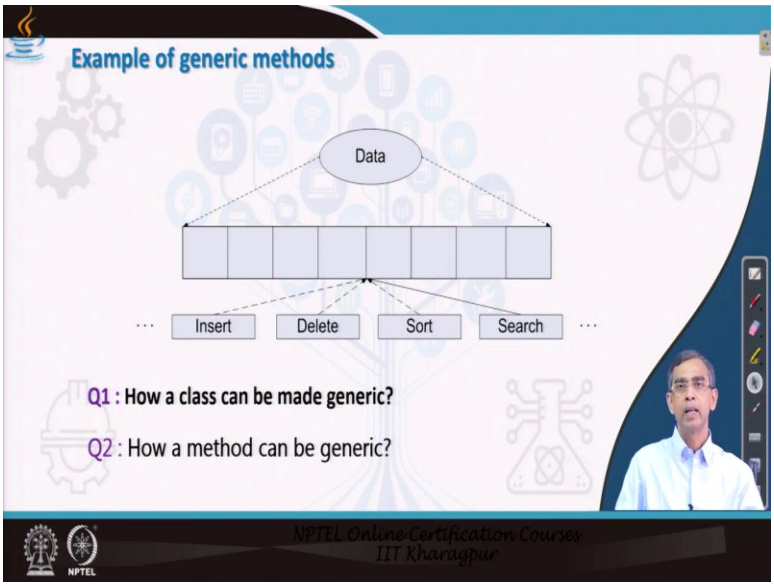
(Refer Slide Time: 4:07)



So this way the concept of generic data is coming into the field. Now so far this Java programming is concerned a program in Java as you know is basically writing class. A class consists of two things is a fields, they are basically data of this program and the methods basically to procedure, to manipulate those data. So these two things are encapsulated, so concept of class that means writing a program is a matter of encapsulation.

So that concept you know already, but here we basically want to think about again this encapsulation concept in a better way that how a class can be designed or you can say program can be written, so that irrespective of type of data your program can cope with. This means how it can be generic, so a class can be made generic, a method can be made generic, where data can be of any type, so this is the concept that is coming on the way so far generic programming is concerned.

(Refer Slide Time: 5:17)

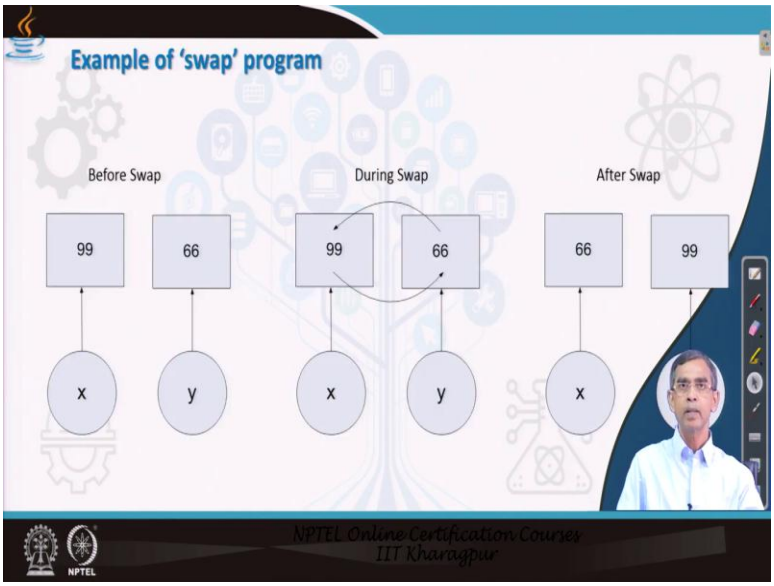


I can give an example, if you want to store an array of data, you can store using array that probably you know, array is basically the collection. Now the different operations those are possible to deal with this data, I have mentioned here how insert and new data to it, how delete a data from it, how sort the data or how search a particular data. Now here if we see, we can write different methods to perform all these operations, insert, delete, sort and searching.

But those methods, again we have to rewrite those method is basically depending on type of data. Now instead of if we write one piece of code that is insert and whatever be the type of data that array can hold, absolutely it should not be an issue, it can work on that. So this is a concept is that how we can make a method generic. Now other than this generic there is one concept is that how we can define an array, which can store any type of data.

So that means generic. Same array name you can define, say name of the array is X, but time to time it can store an integer, it can store a string, it can store a floating point values or any user-defined type, so if we can, then we can say that this is a generic collection or that means a class can be made generic who is basically store the data irrespective of their type, it can process the data irrespective of their type, so this concept is basically comes to the idea about generic methods in C program, in Java programming.

(Refer Slide Time: 6:59)



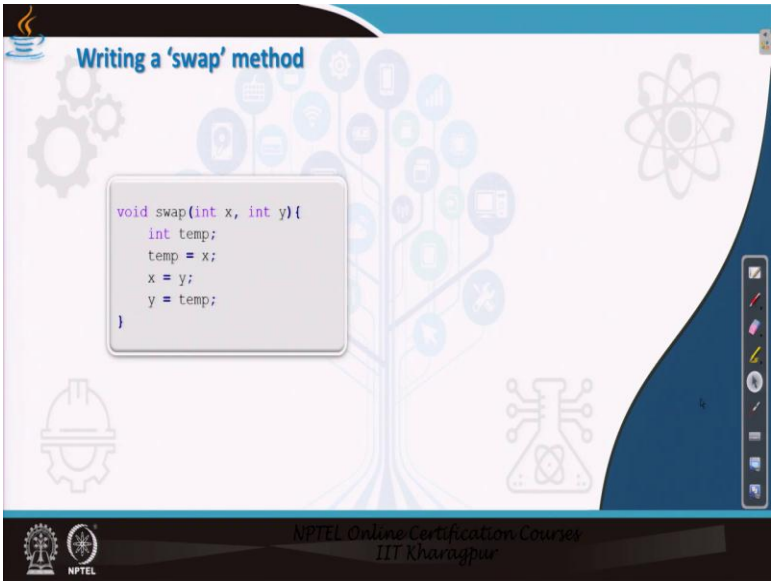
Now, I will like to highlights the concept of genetic programming with a very simple one example and this example related to swap method, it is called a swap operation. Swap method is very simple. You have to define two values, x and y let it be, and they will be stored some value like say in this example X stores 99, y stores 66.

Now, what is the swap operation is that the value of x and y who is the store interchange themselves, this means that X will store the values of 66, whereas y stores 99. Now, this is the very simple one program, which we can write maybe to three lines of code is required to do this

operation. Now operation is very simple, but the type of data that needs to be swapped may be different.

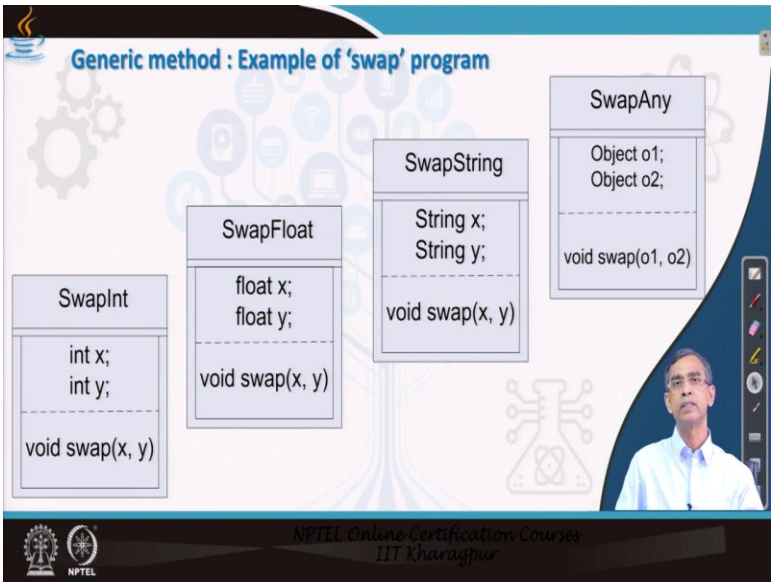
For example, instead of integer data if I want to store some floating-point values, then definitely the same method will not work for you because whenever you want to run a method, which is defined for integer, but for floating-point it will give a compilation error or runtime error, so that is why you cannot. So, what you have to do is that you have to define the different programs depending on the type of data.

(Refer Slide Time: 8:30)



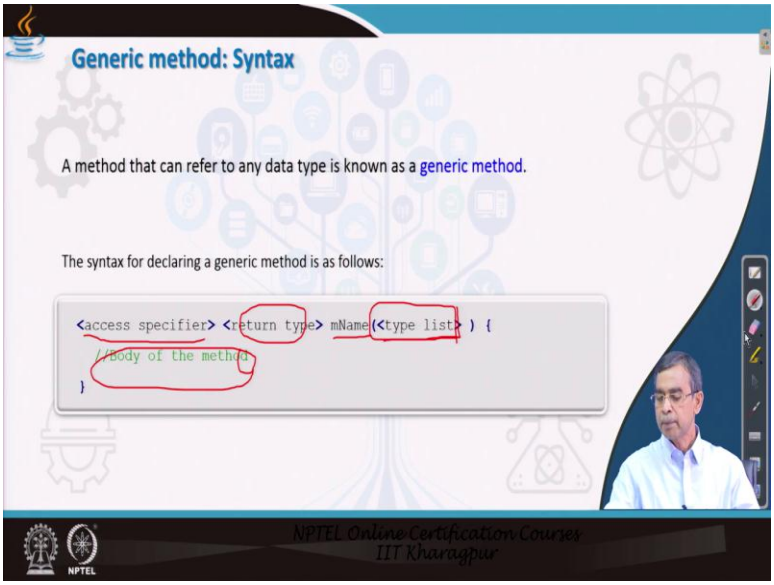
Now here is a simple example about how the swap method you can define in order to do the interchanging the values between x and y. Very simple code but it will only work for int X and int Y that means this method will be called for only integer arguments. If you call these values for say 2.3 as X and 4.5 as Y, it will show an error during the execution.

(Refer Slide Time: 8:56)



So what is the way out? Way out is that the same method needs to be duplicated that mean we have to write void swap X, Y for integer, then void swap X, Y for floating-point values, or for void swap X, Y for string values, or any other user defined type o1, o2 like. So, same logic, same steps, same set of codes, but we have to replicate it for the different type of data.

(Refer Slide Time: 9:27)



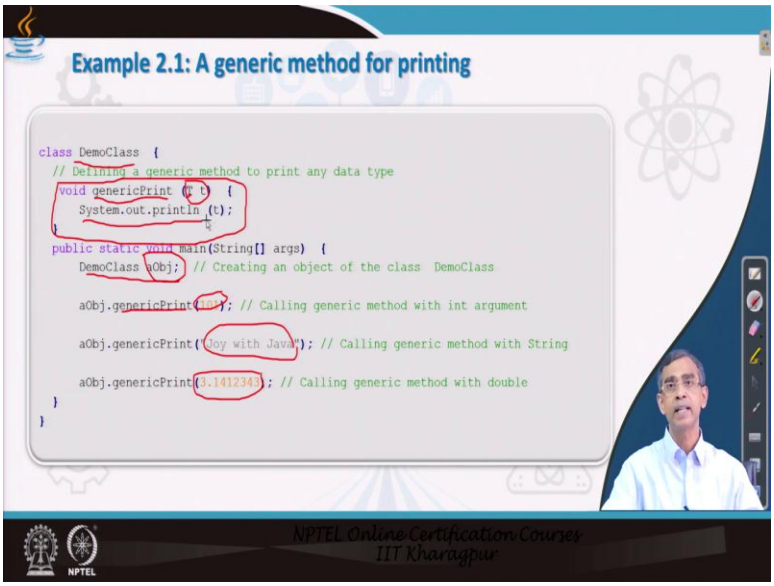
Definitely this is not a good practice to do this because at the time of writing program you may not know what type of data that you have to process or you have to swap for example. So without knowing this what type of data, if you want to write a method and your method can be capable of handling any type of data that will be the very good idea to do that. So this basically concept to define a generic method, Java provides support for defining a generic method.

I have listed here is the syntax by which the generic method can be defined and here you can see this is the 'access specifier' as it is there in every programming languages either public, private, protected, or void, and then 'return type', it is basically the type of value that the method is return is obvious in our concept. Then this is the name of the method by which we want to declare a method.

For example, swap and then here is the syntax that you should note it very carefully. It tells that what should be the types that this method can handle, so list of type. That means you can make a type a generic or a type a specific, specific means it always take only one type of, specific type of values, on the other hand generic.

So they are basically a the syntax is given there, so that you can mention what are the different type of arguments that you want to pass and then what are the variable arguments also you can define to the pass. And then finally this is the body of the method that you can define, that means it basically the different steps is there. Now, we will discussed about how this concept, that how the arguments in a method can be declared in a generic way.

(Refer Slide Time: 11:28)



Now, here is a very simple program for the first timers. I want to give how a method can be made generic and here you can see I have taken and, okay there is an idea about defining a simple program, the name of the class I have given DemoClass. It basically declare one method is a very simple method and at the first loop it is a generic method in our case. I give the return type is void, is a default access specifier is public and the name of the method is genericPrint.

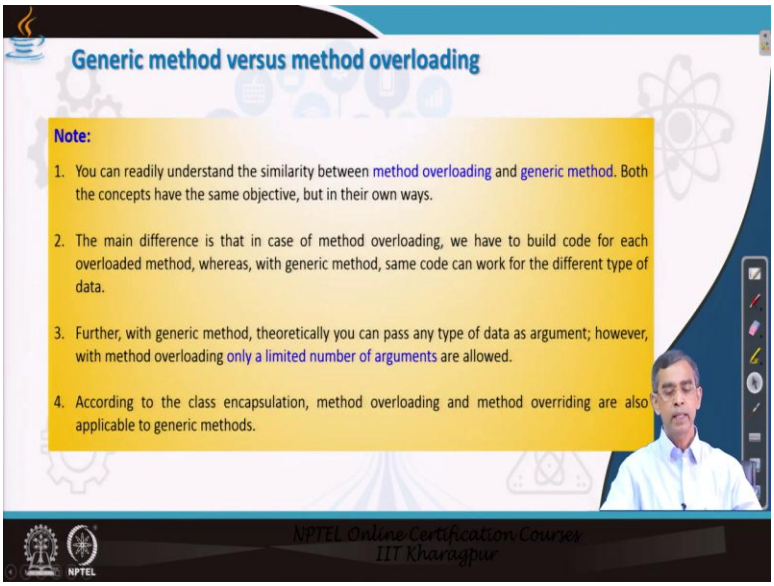
So it basically tell about how a print can be made and this method actually plan is to print any type of data, then the type is basically, it will take only one type of argument and here I can just mention as it is not a specific argument, I mention it is called the template argument, so usually it is denoted by T or capital U or capital V, so in this case I have mention here capital T. So capital T indicates that is a template that means it is for any type.

For example, it can be int, it can be float, it can be double, it can be string, like this, and T is basically the type of value where declaration of the, T is a type of capital T, and then this is the one simple statement that print statement to print the value T, which will pass to this one. Now let us see I create an object of this class that is the instantiation to create an objects and call this generic print method, the generic method, in this case, for this object aobj.

And first call is basically with passing and value integer and the second call we pass the value to this method is a string and the third call is basically called this method with a floating point

value. Now as you can noticed here, I declared only one instance of the method with this template declaration called a generic declaration and then the method can be called for any type of data and therefore, it will work for you. So this is a very simple example which basically explains how the generic method can be declared, a very simple example to discuss it.

(Refer Slide Time: 13:47)



This example you can see that generic method declaration is not a very big headache and it is really as simple as this one only the thing is that instead of regular type that you decide, instead of regular type you just define a template type like it is a T. So, int X, do not write, if you write int X, then this method will work only for integer, but if you write T X, then it will work for you any values X of any type that is what the concept it is called the generic method.

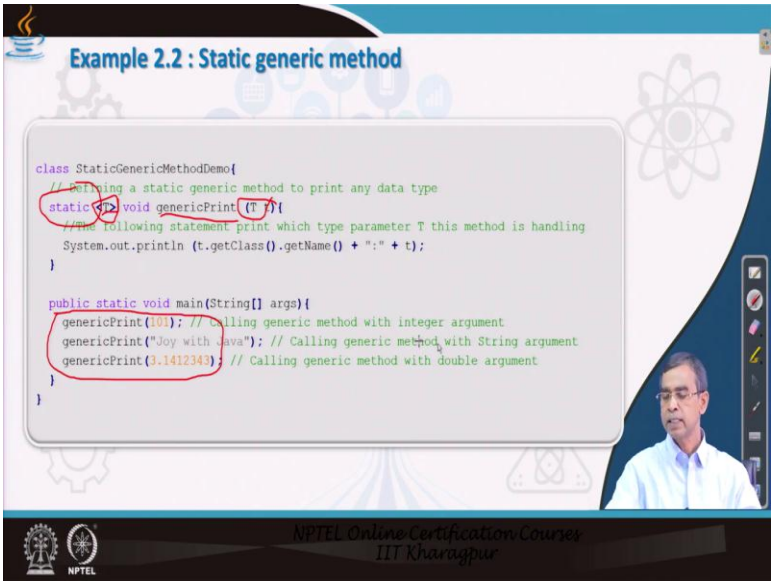
Now the concept of generic method as you see and (())(14:24) possibly you know about the concept of inheritance in your the core programming of Java overloading, so both the things are basically looks alike, actually they are same, but it is more compact way of defining a method and overloading. Overloading basically the thing is that you have to define the methods, which have the same name but for the different argument lists or different return type.

Inside body code may be different, but here only one instance of the code that means only one copy of the code, but it, this copy have the same logic but for the different types of things are there. If you want to follow the different logic, but with different values, but the name of the

methods same, then again you can have the overloading, so generic method, overloading, and combining both overloading and generic method, all are possible and in the same footing as you have learned about it.

So, there is another difference is that in case of traditional overloading method only a limited number of arguments that you can passed, but using the help of generic method, the number of arguments can be anything. So in our next discussion we will going to discuss about how the variable list of arguments that we can passed.

(Refer Slide Time: 15:50)

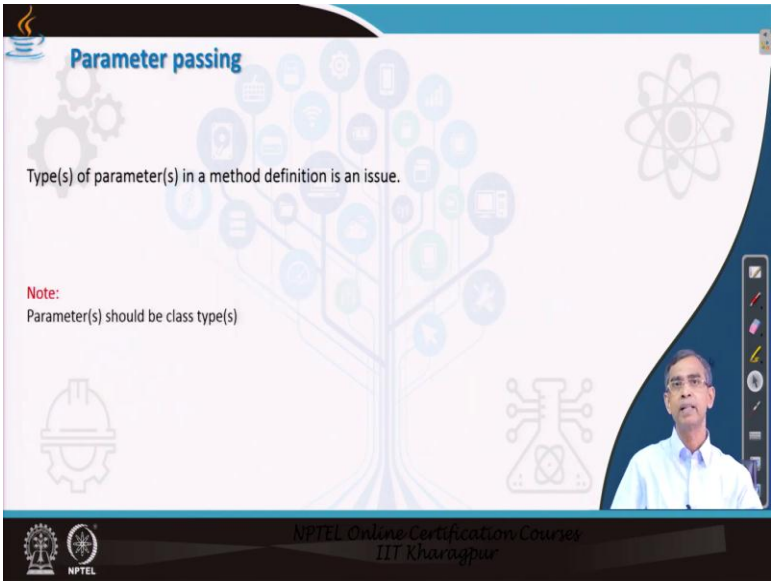


Now before going to that discussion I again want to mention one more in concept like the conventional methods in inheritance or encapsulation, here the generic method also can be overloaded, at the same time it can be made as a static. So here is an example, which basically gives an idea about how a generic method can be made declare a static.

So here we can just use the static keyword, and then this is the return that it can return any type of value so it is discussed generically and this is the name of the method and this is a generic declaration of the method. So this way it can be done, so that means a method can be made static. If we can make a method static that means without any creation of object you can call this method, you can recall the example 2.1 where we created an object and for that object we call the generic method, for example, generic print.

But here you can see the methods are invoked without any creation of object that mean method can be called as a standalone. So this is a way the generic method that it can give you and you can have it and then you can realize it. So generic method, the concept of overloading and then static everything is possible and that we have discussed here.

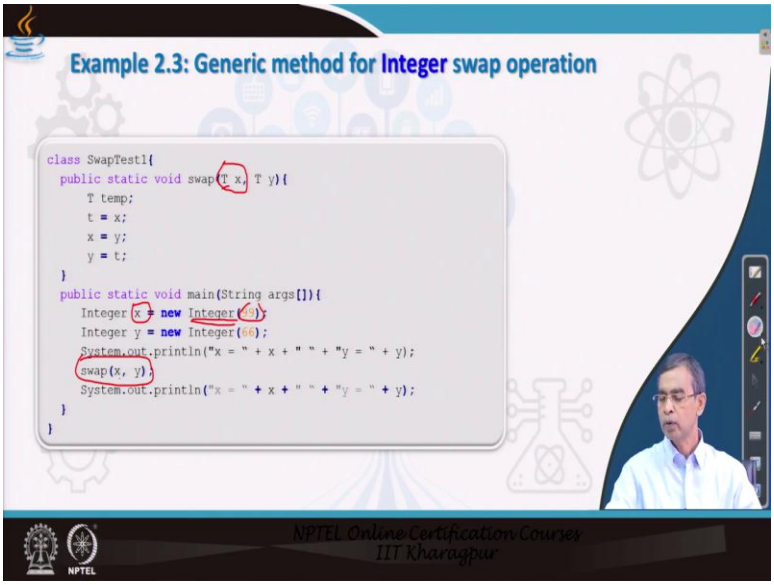
(Refer Slide Time: 17:23)



Now, I will come to the discussion about parameters passing. Parameter passing as I told you, if you want to develop a generic method, which not only takes the template value that means general values, irrespective to a specific value, rather it also allow you to pass any number of

arguments. So it is called the parameter passing using generic facilities. It is apparently appears as an issue, but really in the, with the help of Java programming it is not, it is very simple job, in fact.

(Refer Slide Time: 17:57)



Let us see what is the simple task that we can do for passing a parameter. Now here I just want again another quick recapitulation of the different swap operation for different values, what I want to say here, we can define a method, the swap method generically, so this is basically the generic declaration of a method swap. And once I do it then I can call this method that means this method, the swap method for any type of value.

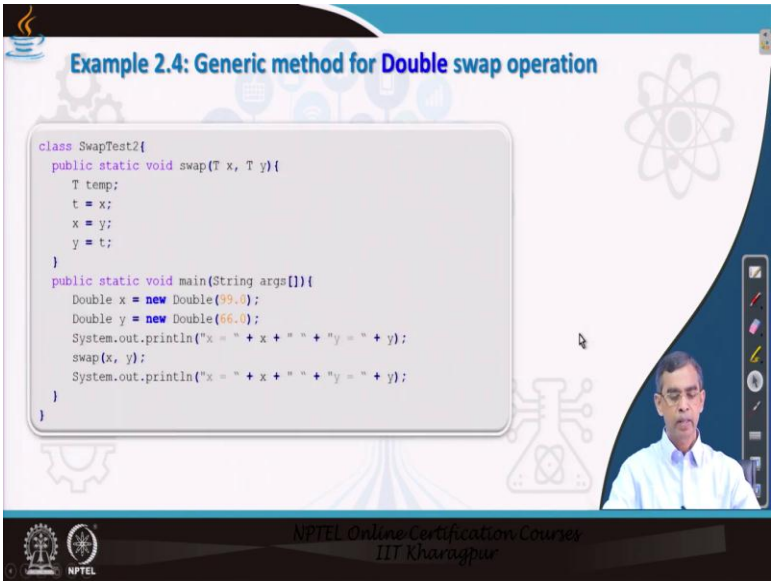
And here you can see the swap method with two parameters instead of single parameter that we have discussed in the last example, x and y, both are of same type, TT, that is why if X is one type and Y is another type, then if you write T and U or whatever it is there, so that is a concept that you have to think about. Now, so this is the idea about the swap is there, now once it is there, now you can see how the method can be called for the different type of objects.

But here little bit different tricks needs to be followed that is why I have just mentioned these examples for you, one thing is that if you want to pass an integer you should not pass simple integer value, rather integer object value by means of upper class, so basically this 99 as we have

mentioned is basically is a integer class object, that means it is an ET, basically an integer value, int value, we can say can be passed to an object like.

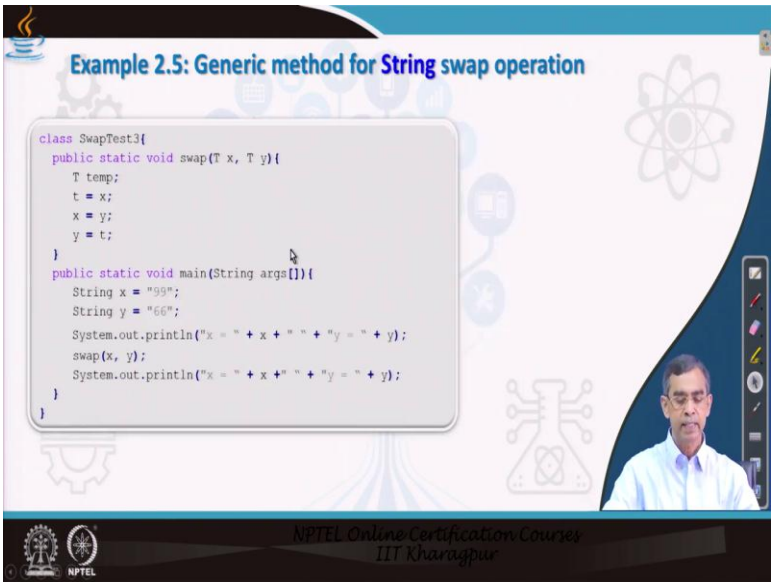
So this object is of type class integer and X is basically is an object of type integer, so whenever we pass it, because in this parameter or T, this basically is not a primitive data like int float double cap, is basically is an object type should be that is why we convert this integer value to its class type and then pass this one, for example, this swap X, Y, here you see X is an object of type integer, Y is also another object of type integer, then call the swap. And this way this method will work for you, this method will work for you, and then it will basically run it. Now this way you once you declare the generic method and the several main or several program, you can call them to pass the different values but at the form of their objects.

(Refer Slide Time: 20:29)



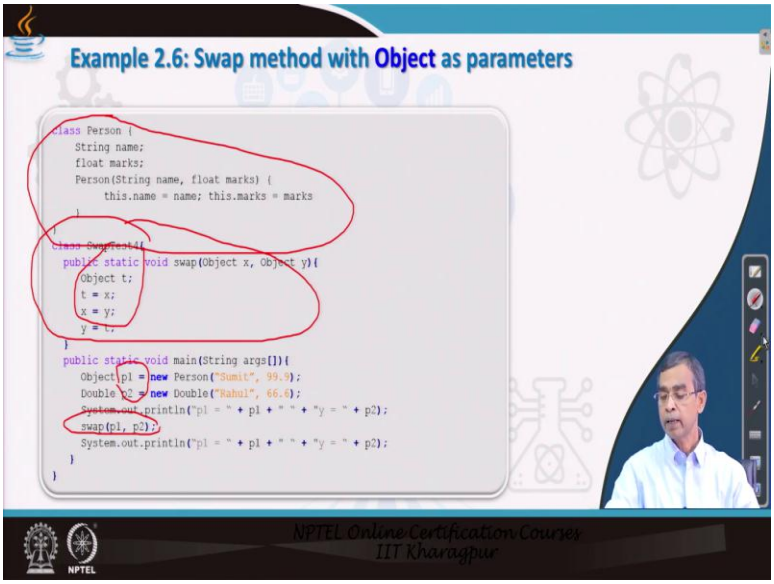
For example, here as we see, here this is the next version of this, the same program, the method is same, but we call this for the double. So a double value which is in our regular sense can be converted a double object and then swap method can be called.

(Refer Slide Time: 20:44)



Similarly, the string, string is an object always, so string, this is the string object, pass it, and then this method will call for the sting values.

(Refer Slide Time: 20:54)

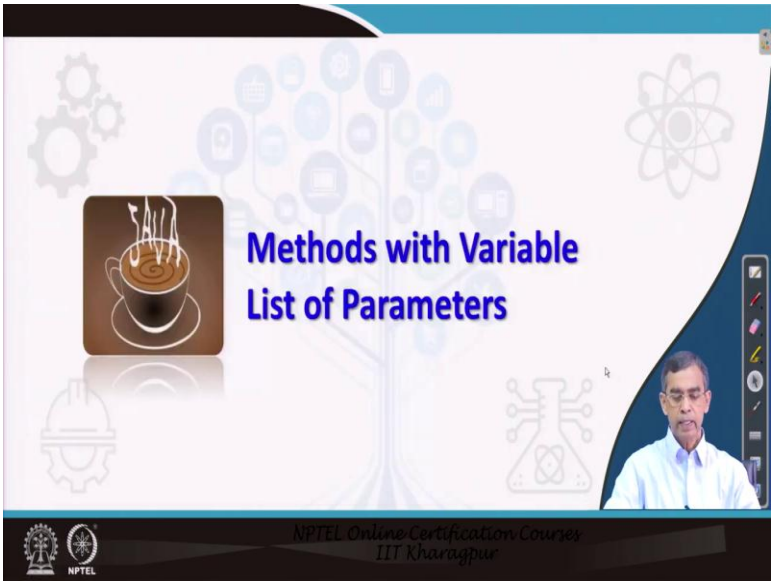


So we have discussed about the three different methods where one declaration the generic method and it can pass the different values, but they are in the form of their object representation like integer object, double object, string object. The same method again can be used for any user-

defined type also. Now here is an example we declare one, we declare one class to define a user-defined type is a person.

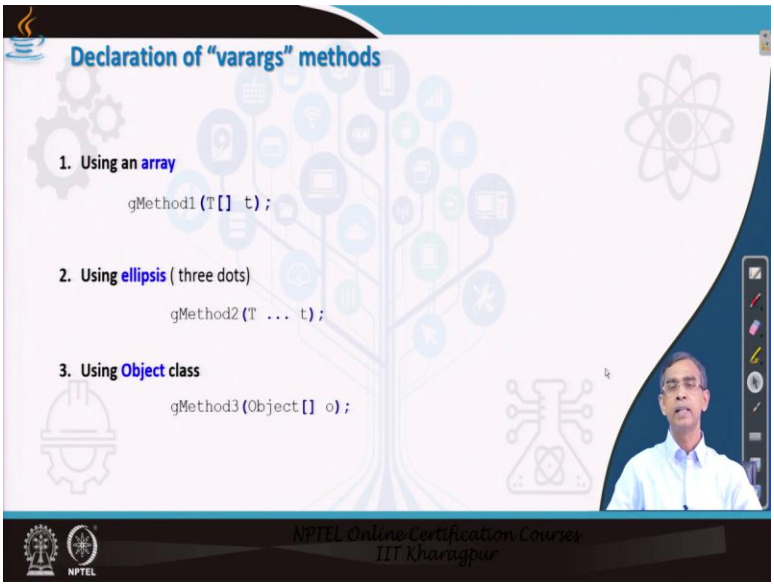
Person has the different fields and a method is a constructor. Now, so this is basically the swap, as usual it is the previous one, the generic method, and then we can create two objects of class P person, namely p1 and p2, so two objects are created and here we call the swap method for the two objects p1 and p2. Now you see the concept is same generic and then call is same, but it can work. It can work irrespective of the type, either integer or double string or user defined type and this is the concept. This is a concept of the generic method that we have.

(Refer Slide Time: 22:09)



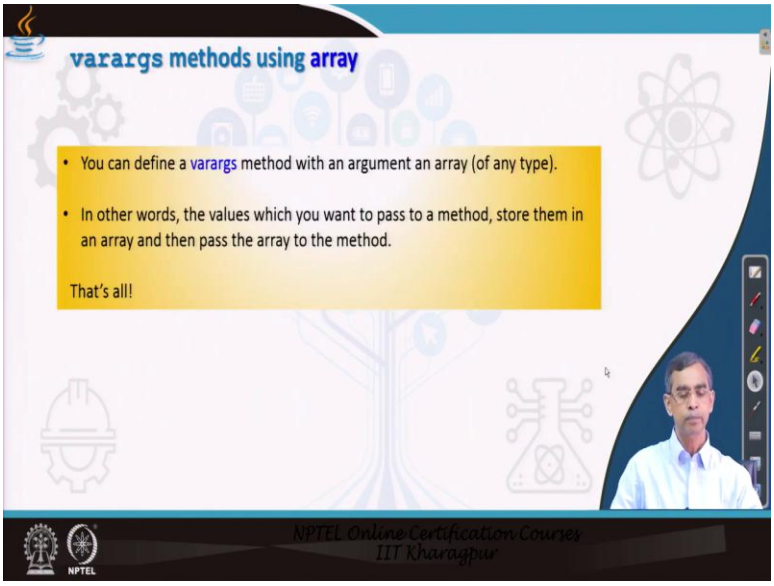
So this is the concept of generic method declaration as we have learned about. I will quickly come to the discussion about Methods with Variable List of Parameters. We will learn about how a method can be declared generic and how this method can be called, whenever a generic method will be called it should be passed as an argument of object type only, that needs to be remember that is all.

(Refer Slide Time: 22:35)



Now, here there are three different ways by which a method can be defined so that this method can take variable list of arguments that mean argument sometime nil, argument sometime ten, arguments sometime any numbers. There are three broad ways that can be done, one is using an array, another approach is called using concept ellipsis, ellipses, basically three dots together, another using the object class.

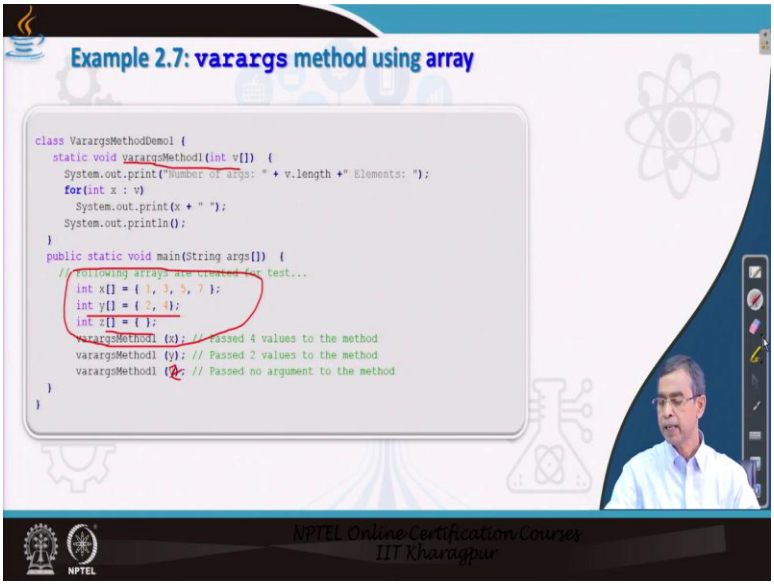
(Refer Slide Time: 23:03)



Now, let us discuss one by one. Now, this concept of defining a generic method with variable number of arguments to be passed is called the varargs method. So it is a (())(23:13) concept in java developer give C is called the simple varargs. So, varargs method with any type that is the generic. How it can be integer array? Now, if you declare an array of any type as an argument, then that is enough, because if the array can store five values, then you can pass five parameters.

If the array does not store any value, null, then it does not pass any arguments. If the array passes hundred values, then it can work for the hundred arguments, so this way using array we can declare a generic method, which allow you to pass variable number arguments.

(Refer Slide Time: 23:51)

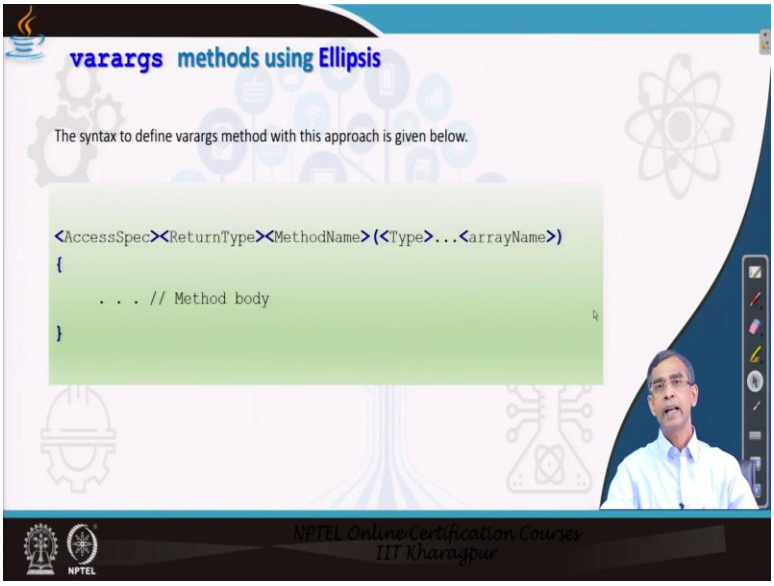


Now, here is an example you just quickly check it and this example is defined a method and this method is basically declared a simple method that varargs method and here we declare int V that means it will pass any number of arguments as an integer. So you can declare at this one. If you want to pass any type of value, then instead of int you can write capital T, then V.

That means the array of any type that means any number of types of any number of values can be passed that is the things, but in this example we only consider, we only consider only say suppose the method is planned to pass integer arguments and any number of arguments for the simple simplicity in the discussion.

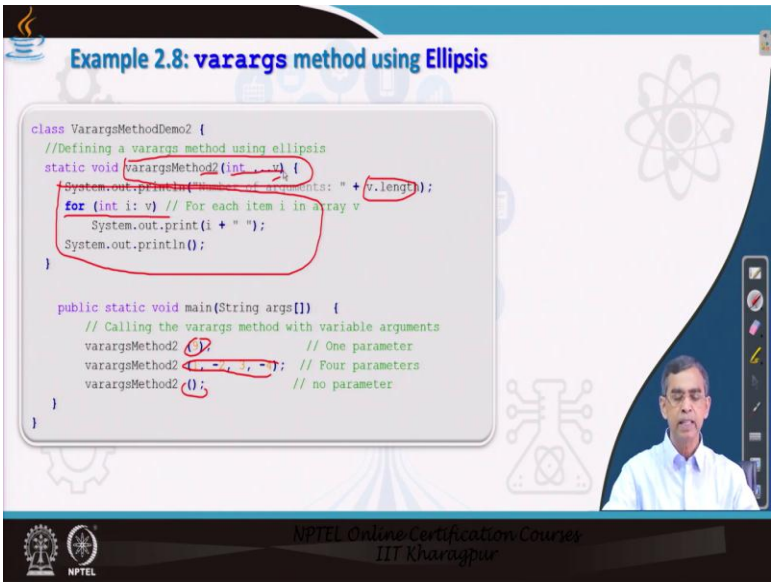
Now, here this is a very simple main method that we have discussed here, in this main method we declare three arrays, so X is an integer array initialized with three values, 1, 3, 5, 7. Y is another 2, 4, Z is another array, which does not include any value, so three arrays are declared. Now, if we call this method, there are methods one for X, this means we are calling this method with four different arguments 1, 3, 5, 7, passed to it. On the other hand, Y is basically called with two values and this is another example, it is Z is called with null values, so this way the different type of, different number of arguments that can be passed to the methods and this is the way that the generic method works for using a simple array.

(Refer Slide Time: 25:48)



Now next step, next procedure is basically using ellipses. So it is a defining an alternative approach, it is better approach than the previous one. I will tell you exactly what is the syntax that we can follow, this is the as usual same up to this one, MehodName and this can be a generic type or a specific type according to your own, it can be integer or it can be generic type. And here you see this is the three dots, these basically called ellipses. And then this is a name of the array, which array basically you want to pass it to that. Now if you do it, then what it basically tell that it will allow you to pass any type of values to this one, instead of simple array type it is there. Now, let us have an example to clear this concept.

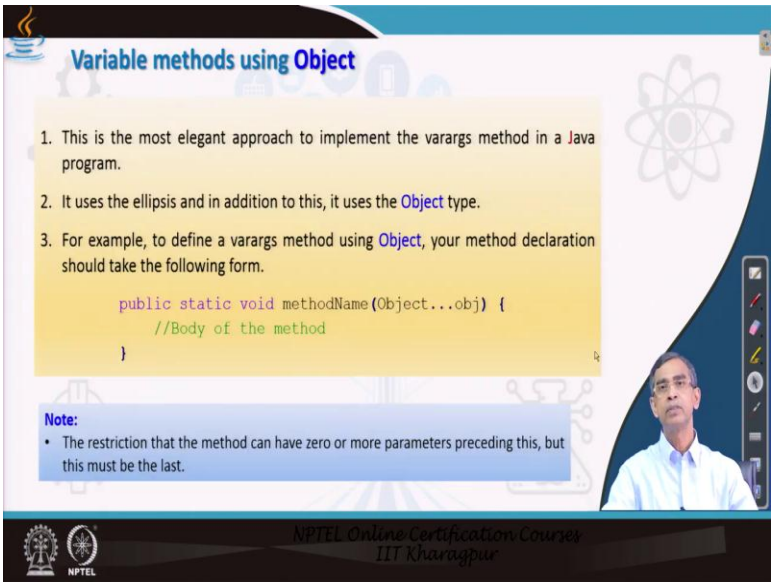
(Refer Slide Time: 26:45)



So this is an example that we show you, yeah this is an example that we can see, so this basically the similar way, the earlier one we have done it and this is the varargs methods and where we can show that integer V, then here we can just have a simple statement of this basically printing an array of values that we will pass to it and then V dot length is the another value that we can do it and it basically print it.

So, is very simple, we will pass an array and in this array we will basically, in this method the array element will be print easing for each statement that it is there, now we can call the method here, for example, the method is called call single as it is the array and here is the four and this is the null, no parameter passed. It is basically similar to the previous one using array, but it basically use the concept ellipsis, that is the only syntax that it matters here, but it works the same way than the previous one. So this is the one, the second approach is called the ellipsis.

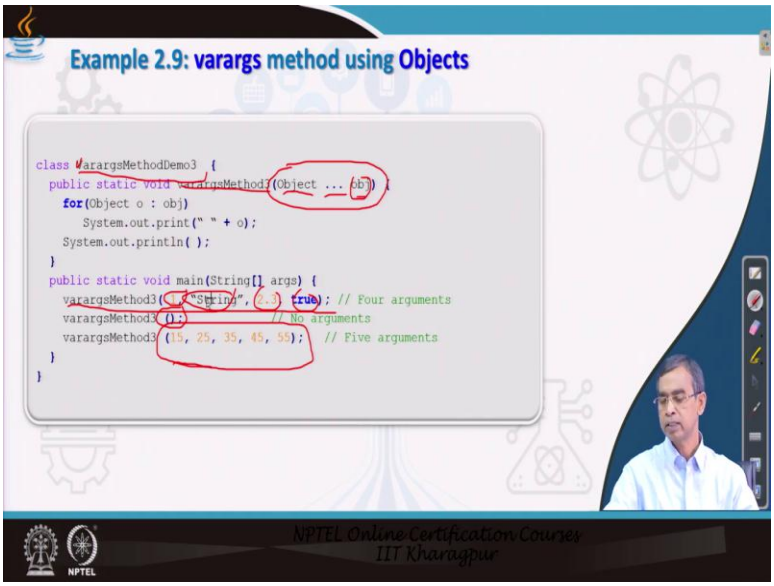
(Refer Slide Time: 27:59)



And the third approach, the third approach using object. It is more versatile, more better, now you know the object is a type, here object is a type is a superclass of any objects. In the concept of inheritance actually object class is the top of all the class that mean it is the super of any child class actually.

It means that if you pass any type of object, it basically you can pass it at the type of object, because object is the superclass of any object, so it basically holds any type, which derives from this, so this concept is followed here and here is the idea about, just I can say here using object type we can declare and the ellipses that mean any type of values and with any number of parameters, you can pass it to there. So this is the concept that we can follow that using this object declare definition we can pass it and then pass any number of values, any number of arguments.

(Refer Slide Time: 29:21)

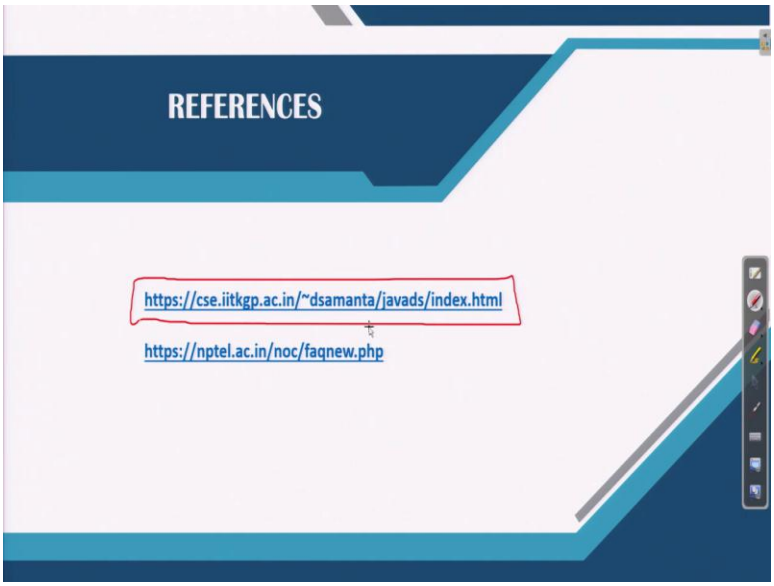


So this concept again can be illustrated with a simple example, yeah, this is an a simple example here so likewise the previous one we declare the varargs methods, and here we declare using objects, so you can note the difference is that we declare the type of objects, this is a elliptic that we have discussed, is a object, so that means we can pass any type of values that we can and any number.

Now here is an example we can see, we call this method the varargs, so it is actually small, the varargsMethod3, we call and here we are calling this method. In this call we pass integer, it is a sting, it is a float, and Boolean. So that means this method, in this call we call for four different arguments of any type and using this declaration it is possible, because object will takes any numbers as well as any types.

This method here we call only for the no parameter is passed, and here the method is called all parameters in case of five parameters are passed as an argument all are of same type. So this is a quite versatile concept that we see using this, here a method can be declared in a generic way and it can be called for any number of arguments, any type of arguments.

(Refer Slide Time: 31:02)



Fine, so these are the different link that you can follow for your further studies. This is the link. I have already uploaded all the concept, information and program that we have discussed in this course, you can consult this link, you can find in this, HTML file is there. Further for your, further any doubts or any clarification, any queries you can write mail to me as well as you can write, post your problem in the discussion forum, so that discussion forum you can have during the run of the course. Thank you very much.