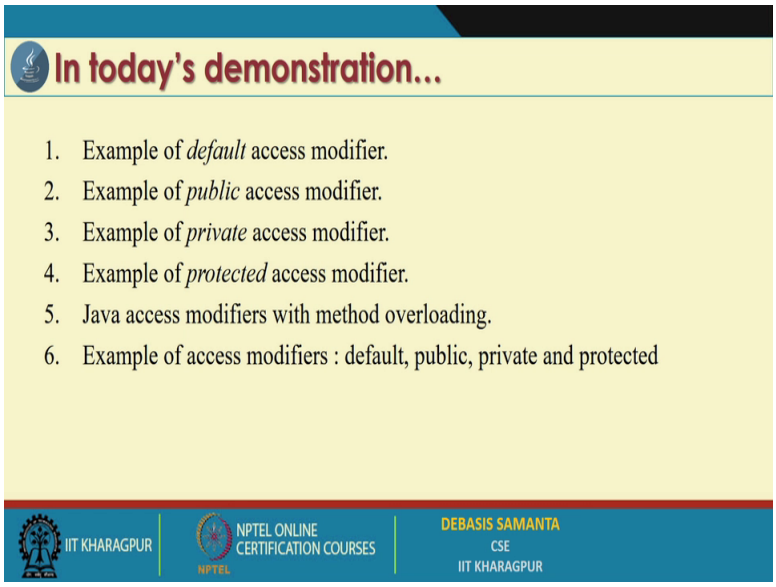
Programming in Java
Prof. Debasis Samanta
Department of Computer Science Engineering
Indian Institute of Technology, Kharagpur

Lecture – 16
Demonstration - VII

So, information hiding is a very important and object oriented paradigm and information hiding is also very nicely has been featured in Java program. So, we have discussed about the information hiding in our last module. Now, it is our time to have a quick demo of information hiding concepts in Java.

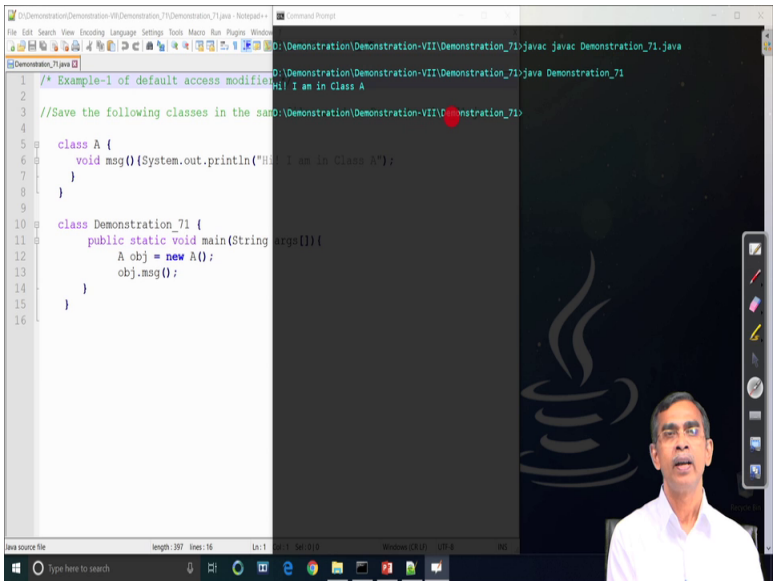
(Refer Slide Time: 00:43)



As we have already discussed at that information hiding in Java is by means of four access specifier; default, public, private and protected. In this demonstration we will demonstrate how the different access modifier will have the different access protection for the different classes in Java program.

And then with this modifier the method overriding also method overloading as well as overriding how the two things can be there and finally, we will have a quick demo about having all the modifiers in one program like. So, this is our plan of the demonstration today. Now, let us have the first demonstration in this series where we were going to discuss about usage of default access modifier.

(Refer Slide Time: 01:33)



Let us have the program here so, class A is declared as a default class because, if we do not specify any modifier before a class name then it will be treated as a default. So, class A in this case is default and then you same thing the message void is basically is also a default method because no access specifier is mentioned.

So, as a whole the class A is a default class and with a default method the message. Now next is that main class which is also default class here and it is basically in one file because both the class A as well as demonstration underscore 71 is stored in the same file. So, there is no issue of running it now. So, if we run this program so, it will basically run successfully, but it will run with default access specifier for this program it will run let us run this.

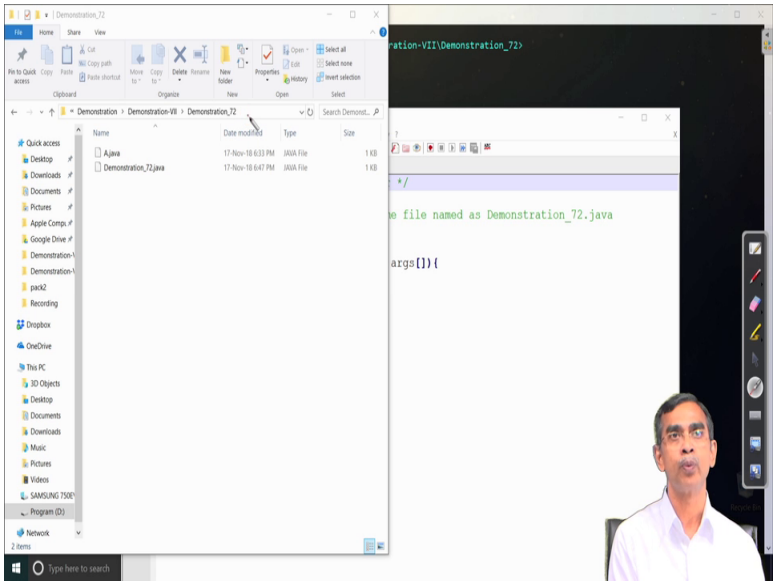
So, as it is compiled successfully definitely execution will be there and it will give the execution. So, it is the successfully it executed now. So, program is not here actually, but program will be seen somewhere else now let us see. So, here basically in one file the in this case we use the name of the program file as demonstration underscore 71 dot java. In one file we store two classes, but for a good practice that a Java programmer should maintain separate file for the different classes.

So, here is an idea about that class A is store not in the same file as in the main class, but store in another separate file and name of the file is same as the class file. So, class A stored in the file call A dot java and demonstration underscore 72 is a new demonstration

showing that two files they are in two different two classes are in two different files if it is there whether, they are if it is here in the class a file if you go to that you can see that this class A is declared here with default access specification.

Now, let us switch to the demonstration underscore 72 program file it basically includes the main file here it is also default. Now, here we can see these two files are stored where we assume that these two files are stored in the same directory here underscore demonstration 72.

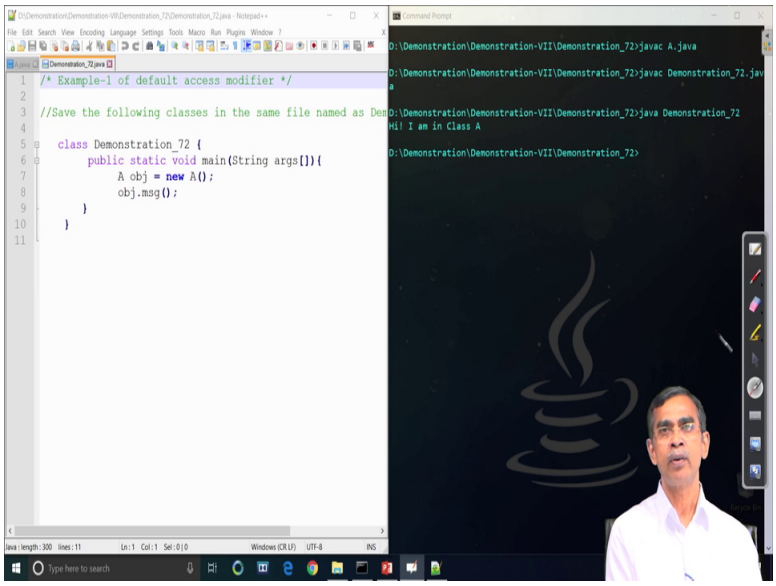
(Refer Slide Time: 04:07)



So, if that two files are stored in the same directory. So, resolution is not a problem and if the files are with access specifier so, member or accessing method is also not a problem. So, here from the main class we are trying to create an object A as A is the default class. So, object creation is not an issue and then we are accessing the method m s g the default method in the class A also not an issue.

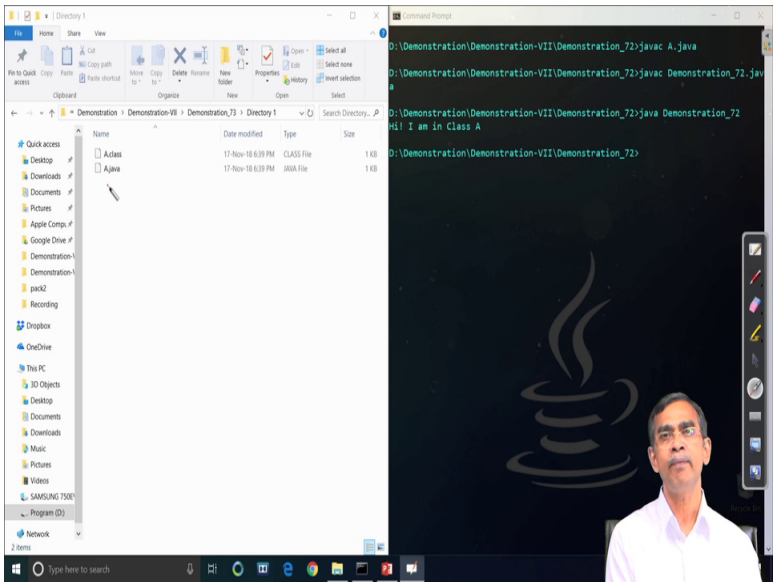
So, in this case if we run the program it will run successfully this means that access specification is access specification allowed to access within the same file or in the same directory. So, no issue if they are in the same file or same directory access modification will work for you. Now, yeah so, this is working.

(Refer Slide Time: 05:01)



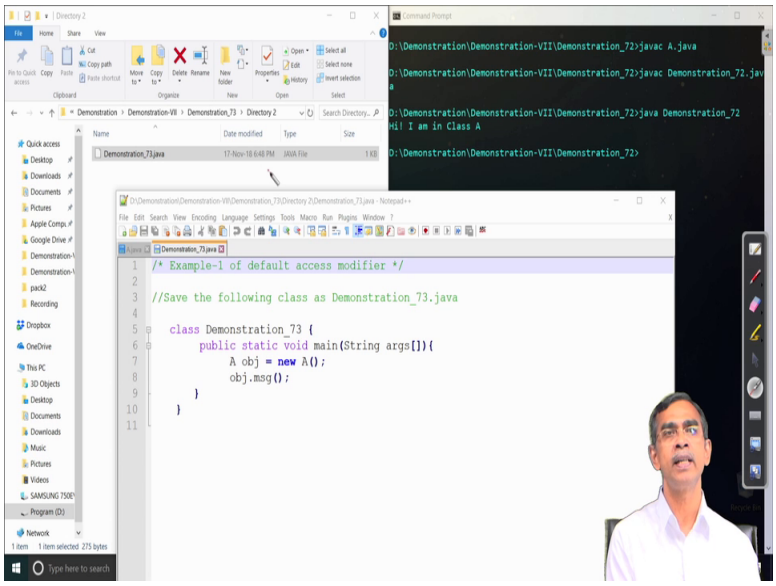
Now, in our next demonstration we will see if the class file A moved to some on directory and then if we can run this program if which is from other directory, but with access specifier is a default. Now, here if you see we created one directory demonstration underscore 73 under we can create a directory 1 where we have stored the file A dot class and A dot java.

(Refer Slide Time: 05:33)



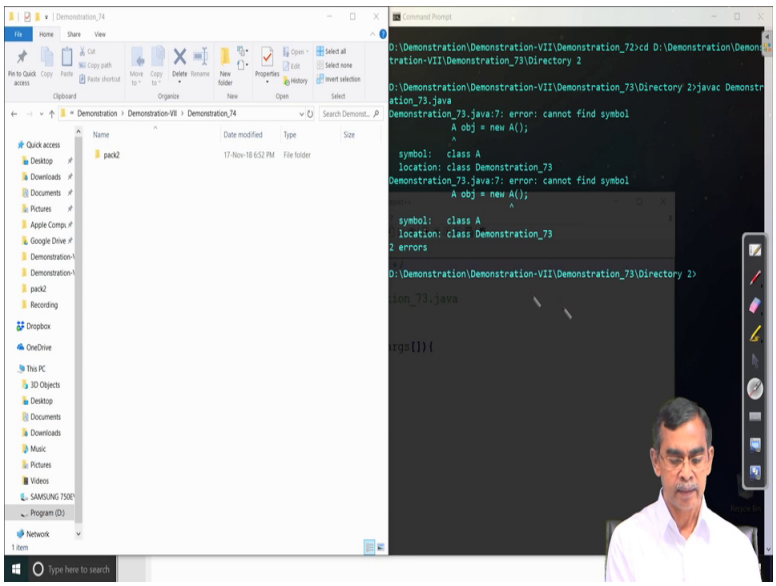
The file it is there A dot java let us see the A dot java file here.

(Refer Slide Time: 05:41)



So, here the same file, but main stored in the directory the different directory. Now, let us go to the main class which is again store in the different directory here, go to the main method yeah. In this case main method is stored in the directory demonstration underscore 73. Now you see the class A dot java file is store in different directory where as the main method is store in different directory. Now we are trying to execute run this program before going let us compile this now see whether it can compile successfully or not.

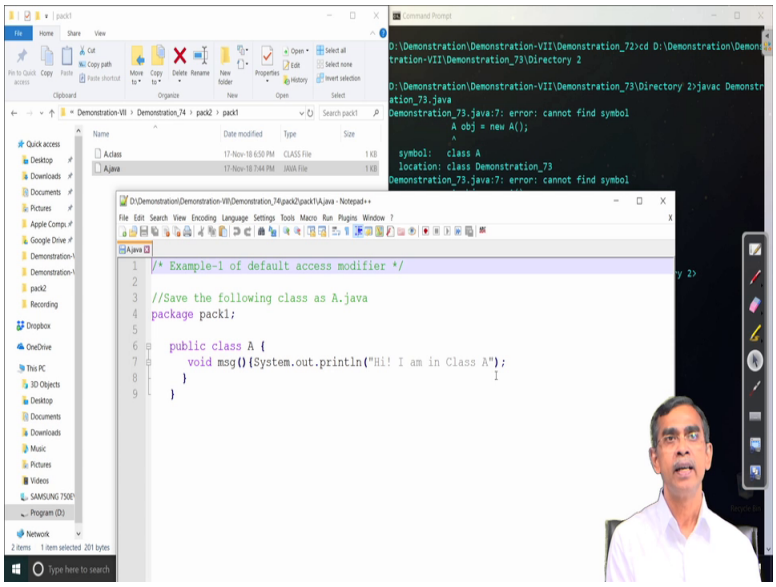
(Refer Slide Time: 06:19)



So, if it is an access specifier is default and the two files are in different sub directories then it will not successfully compiled. Here you can see the compilation error is that that is basically error cannot find symbol A; that means, A is not known to this file. So, access specification if it is default then that is not accessible to any other file outside this directory ok. So, this is why this error is giving there.

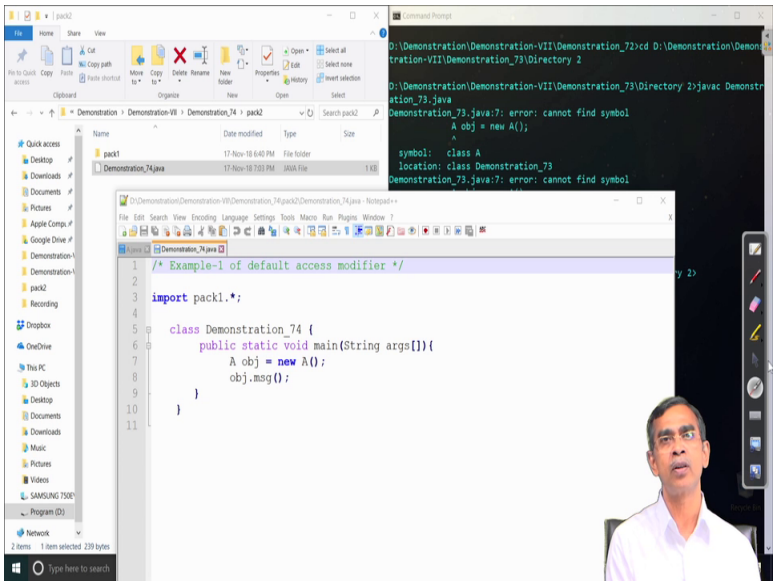
Now, so, this program shows that the last three programs that we have discussed about shows that how the access specification default works for us? Now, again we have some other demo here let us show the 7.4 program here where you can see ok. So, this program is basically creates a the class A file in one directory as you have a directory is also can be termed as a package. So, we create one directory called pack 1 and under this pack 1 we store on file A dot java let us see the A dot java structure here fine.

(Refer Slide Time: 07:33)



A dot java is very simple here we just mention that this A dot java is in package pack 1 actually it is a directory this one. So, class and one thing you can see here we made it public. So, here class A is not access a default access specify default, but it is made as public. Now in the last example that we have shown if it is default then it is not accessible to that, but if it is a public then this class can be accessible outside to any file whichever it is whether the same directory or in different directory. Now, let us have the another program say demonstration program the main class which is stored in pack 2; pack 2 is a another directory and we will define here.

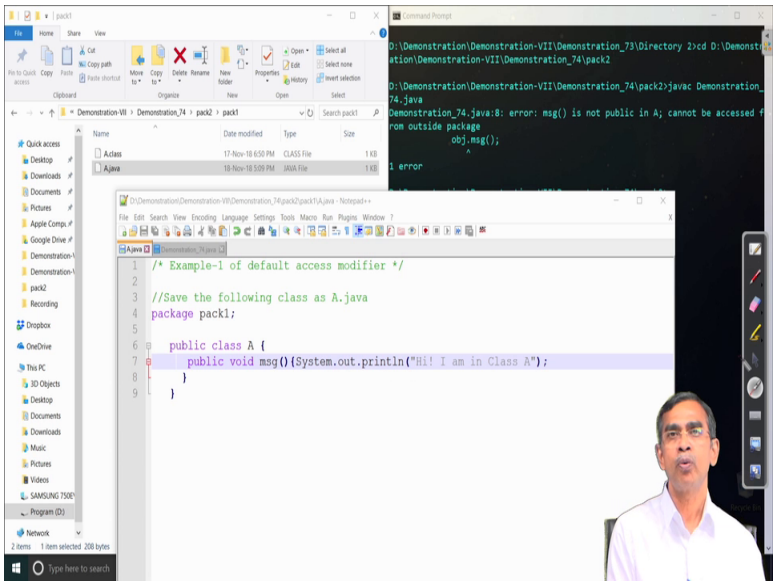
(Refer Slide Time: 08:21)



So, in under this pack 2 we store this program that name as this name is demonstration underscore 74 dot java file. And this is basically we have to give an input pack 1 showing that it has the input capability means, whatever the classes which are declare there in pack 1 directive will be accessible to here because of this regarding this things we learn about whenever you cover the package concept.

Now, here with this if we run this program earlier similar kind of program was not successfully compiled, but here you can see this program is all right it will run for you correctly.

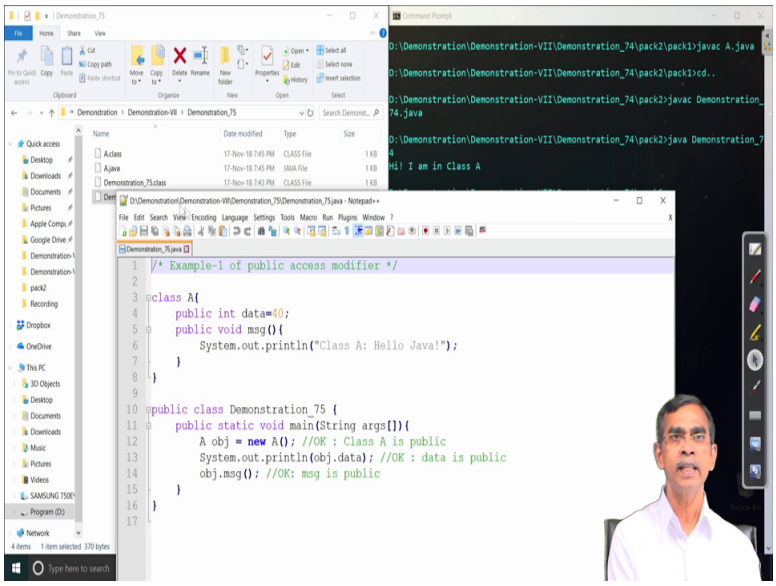
(Refer Slide Time: 09:03)



So, what is a error it is here? Message is not public ok. So, it is not working lets return to the class 1 yeah we cannot access because it is a message is default here. So, if we make it public yeah just make it public then this error is giving there because message although class A is public, but its method is not public. So, this is a public is more weaker restriction than the default 1. So, default is super void here. So, default has the highest privilege this one so, that is why this message is not accessible.

Now, let us again go to the main method it's a compile it so, right yes. So, we have to compile because you have change it and the next go to the main method main class right run this and now it is successful and it is running fine.

(Refer Slide Time: 09:47)

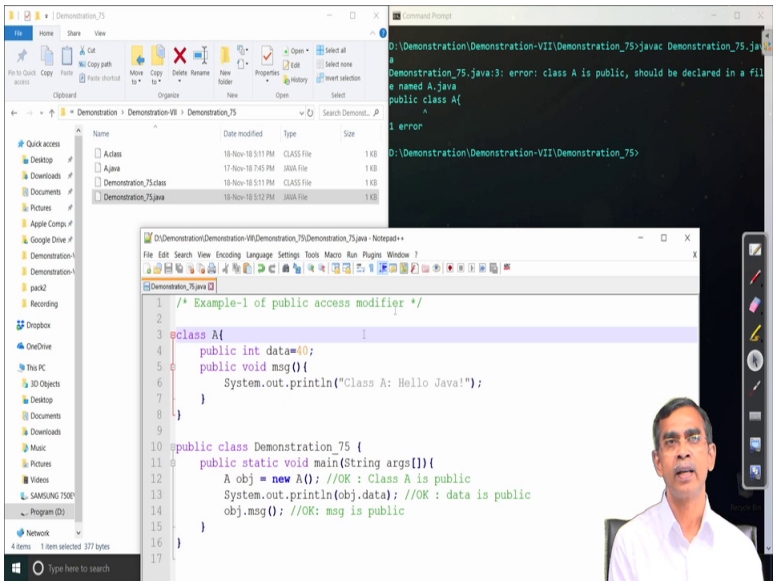


So, this now works for you. So, a public method and with public the public class with its public members is highly accessible from outside any class that is the example explain us. Now, our next example showing on the public access specifier let us have a quick look a another program let us go to the 7.5 dot java here yes. So, hah just load it yeah fine yeah. So, in this program we can see so, we have declared the two classes class A; where the class A is declared as a default and it has members 2 members though both are declare as a public.

So, it public means as we know that this class is accessible to any other class belongs to the same directory; however, this class is not accessible outside to any file belongs to other directories that is fine so, but here this class a as it is in the same file of this demonstration underscore 75. So, no issue we can use it and here we can see we just create an object of this class here and so, object creation is and we also access the method here.

So, public method is accessible anywhere, in the same file, in the different file, but in the same directory or in the different directory in the different file no issue. So, in this case it is no issue. So, if we run this program it will run it correctly. Now, let us compile first this program and after the compilation will execute it.

(Refer Slide Time: 11:51)

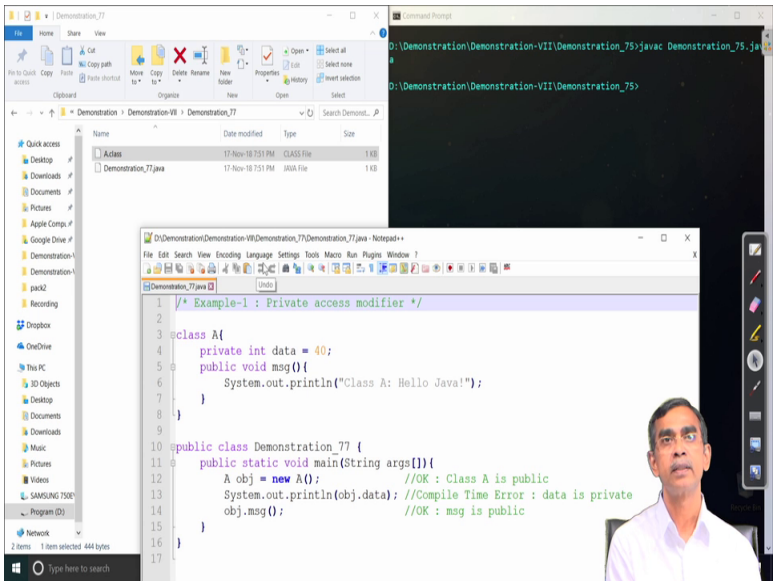


So, that it will a run it yeah so, it is running. So, fine it is running successfully yeah. So, it is running now let us switch to the program again I just want to do little bit modification. So, that we can see it a twist is here. So, A class which is in the same file whether we can declare with some other access modification for example, public let us specify the access for the class A as public ok.

Now, let us see what will happen if we try to run a compile this program what will happen, we made it public and as you know the public is public. So, it can be accessible anywhere, but what will be the problem you can see yeah. So, here you can see here class A is public and should be declare in a file named A dot java; that means, if you declare a class public then that class should be stored separately in a different file name. But, if you want to store in the same file name do not specify any other access specifier other than default. So, it is not required here ok.

So, if you use two or more classes in the same file you should have the access specifier as a default no other access specifier is allowed there, but in the separate file then you can specify any access specifier. So, this is the example that we have discussed about regarding the access specifier and then public specifier and let us have another example about the. So, private access modification let us go to the program 7.7 here we use the private access modification.

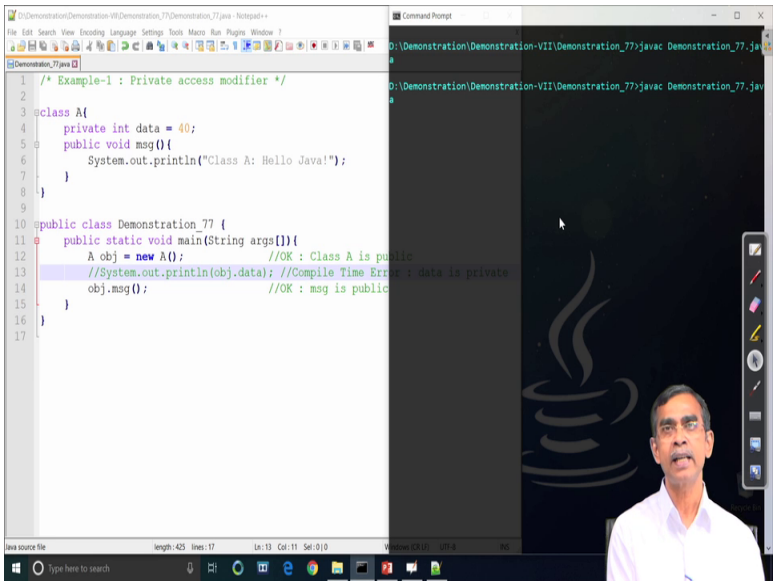
(Refer Slide Time: 13:49)



Now, let us have the program here yeah we can see one class yes ok. So, class A here is declared and it has 2 members; one methods one method and 1 members the data is a 40 as a declare as a private and then message is declare as a public. So, 2 members data and method one is private another is public. So, and then class A is a default with default access specifier.

Now, in the main method main class demonstration underscore 77 is the main class name here we create an object A. So, it is because a is accessible to its own file and then system dot out dot println object dot data. Now here the comment is that compile time error this is obvious because the data which is declare in class A is a private and then private data is not accessible outside this class A it is accessible to the class a itself, but not outside. As the demonstration underscore 7 sees an outside class of class A so, we cannot access so; this is why it will give an error.

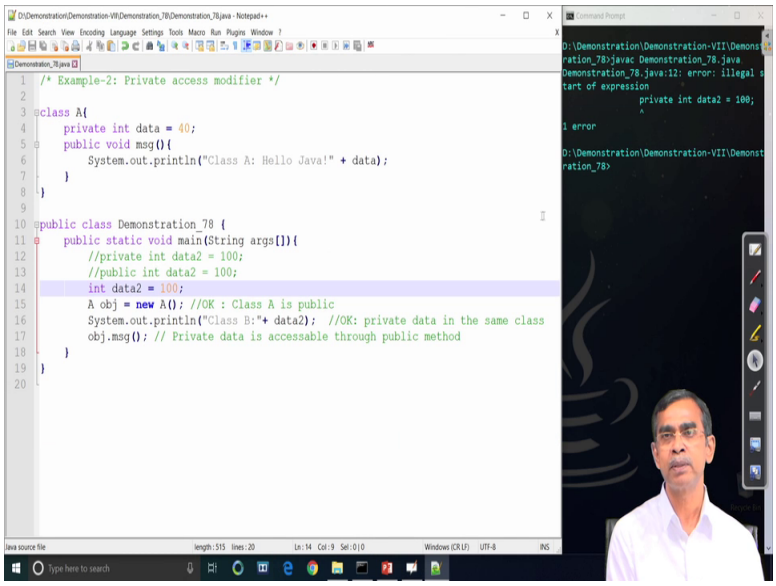
(Refer Slide Time: 15:17)



Now, let us run this program and we see it is giving an error yeah now here you can see error data has private access in A. So, as it private access so, we cannot access it this one if we comment it then definitely it will work go to the program and comment it yeah, comment it yeah fine no this one no system dot out dot println.

So, its comment yeah then it is work it will work because we are not accessing any private members to any other class fine. So, this actually shows that how private members access specifier is work there, let us have one more example about private access specification 7.8 please go to the 7.8. Now, this is an interesting program a program on you just note it.

(Refer Slide Time: 16:11)



It is the same problem as in the last program one, but it is a little bit different is there the same problem in the sense that class A has the same specification private and public a message method. Now, let us come to that main class demonstration underscore 71. Now here this is the private int data 2 1 private member is declared here we can declare a private member here no issue that is fine. And also we declare 1 public member data 2 whatever it is there now fine let us see let us first declare about private int data two uncomment the private one uncomment this one ok.

Now, what we have done here is that in this class main class we declared one main method is there and then one method on data is that which is declared the private and we int data 2 ok, you know private is comment there no that is not correct. So, let us a comment it private comment it private fine now. So, data two is declared as a with access specifier default. So, default and an object A is created here. So, object creation is not a problem because the default class is used here.

Now, system dot out dot println class B data no issue because is a public to this method and obj message is also not an issue because is a public method here. So, here, but here you can see if we run it then see that message m s g is basically access a private data no issue, but it is basically access a private data in directly in main method which is in class other than the class A.

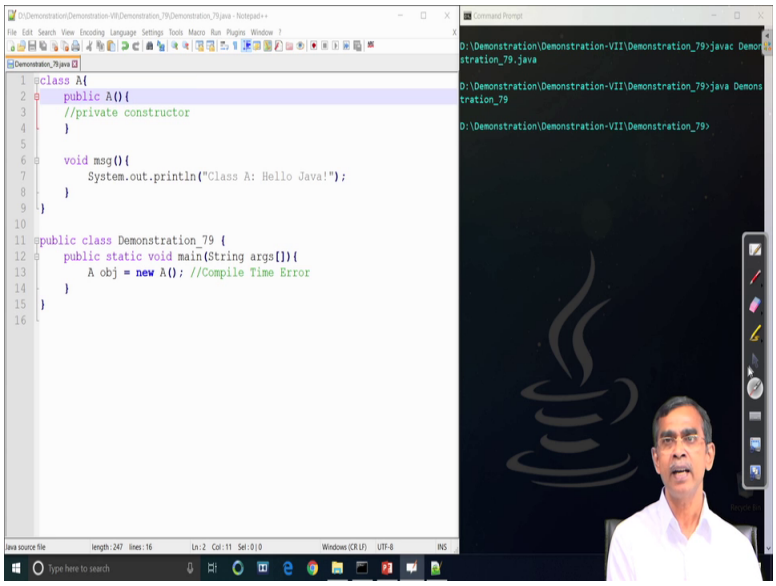
So, this is quite now this is the one important point that you can note it there let us switch to the program again I just repeat it here again here you see message being a public is accessible to any other method. So, we can have the obj dot message in main method. As message is public it accessed, but message this message is even access the private data which is private to class A, but here the as we can object message we can see the result. So, that 40 is now accessible to a method a method which is outside the class A actually ok. So, this is the idea about.

Now, here again let us switch to the program here. Now can we declare a private method in this main method here now let us see integer data 2 may be comment it please comment it and we declare public here ok. So, fine now let us declare here as public and let us see whether we can run this program or not yeah fine now here it is an error you see why the error it is there. So, error is that public int data 2 a things cannot be declare as public only we can declare as the access specifier default because in the same file we cannot declare a method any member as a public. So, public int in this case is giving error as it gives the public as an error a private should give an error also.

So; that means, we cannot declare one member in a main method or is a main class without access specifier any other specifier. So, let us see again private. So, private data now what is the lesson that we have learn from the in demonstration is that in the main class if we declare any member this should be declared with default access specifier. No others access specifier is allowed to declare any member in the main method that is the important thing that you should note it ok.

So, this is the concept about public access specifier and in some sense the private access specification. Now our next demonstration is basically a constructor whether a constructor can be declared as a private or some other access specification other than public.

(Refer Slide Time: 20:37)



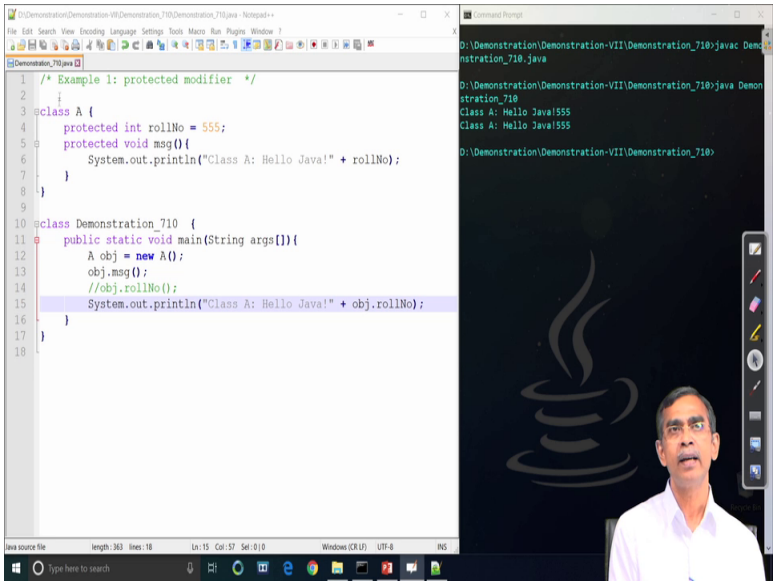
So, our next demonstration 7.9 includes this clarification here we can see one right yes 7.9 here we can declare a class as a default class, but its constructor is declared as a private. Now, we if we declare a constructor as a private what will happen that it will not be able to create any objects in any other class although it is accessible to some other class in the same program may be. So, here in the demonstration underscore 79 here you can see we are trying to create an object A whose constructor is private it is giving compile time error because constructor is private. If it is a private constructor no object can give can be created in any other class.

So, here is the error is a compile time error showing. So, this basically gives an error because a constructor is private if we remove this constructor private may be say a public right here public is are not required fine you can run it then save it and then run it first you have to compile yeah then compilation is successful so, it is running yeah. So, it is running there is no object new oh you do not have any free in method. So, message is there object right obj msg here.

Now, so, we can create an object which was not possible by means of private constructor, but if it is a public it can be accessible here fine. So, this program shows that a constructor cannot be declared as a private in any class if it is there then no object can be created. Now our next demonstration showing the protected access modifier in a class, as

you know the protected access modifier is basically access limitation it limits its access to only the subclasses inherited classes.

(Refer Slide Time: 22:43)



Now, here is an example class regarding the access modification called protected. So, class A is a default but, with its 2 members the protected as an integer roll number and a message as protected message. So, this is the one class here now have the main class say demonstration underscore 710 now so, far object creation is concerned. So, it is not an issue because it is a default class, but accessing the protected message obj msg or accessing a protected member value. So, roll number is a problem, let us run this program as obvious this program should not pass the compilation task step.

So, it will give a compilation error yeah as we see object or roll number it giving say compilation error because cannot find symbol look like this one. So, it is the error is there, but object is created however. And another important thing obj dot msg it did not give any error report actually, if a method is declared as a protected and if it is in the same file then protected is also accessible, but the protected member is not accessible via the this one fine. So, it is not working, but here again we have done some mistake actually.

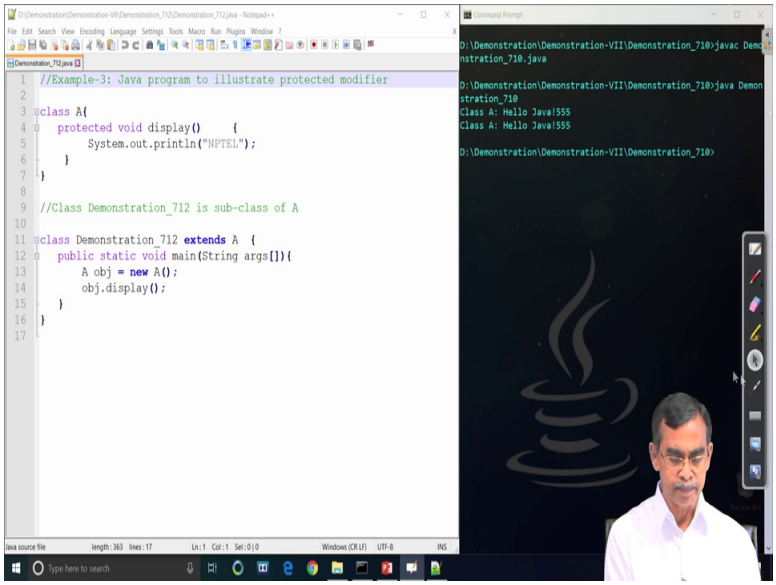
Now, here obj roll number you just see a roll number it is the method you have given not method not method just gives a obj roll number right. System dot out dot println obj roll number just give system that is same statement you can call it there. System dot out dot

println just copy the object object roll number roll number is not a method nah yeah correct you can type it here that is fine in yeah correct here and then commented object or roll number this is not a valid one yeah.

Now, see what we have done here a class A is a default class with protected, but if the class is in the say axis in the same file whatever it is a protected it will be accessible. So, so protected members is accessible in this program in this case this program will not give any error it will compile successfully what is the problem you have not run it successfully compile it. Roll number you object obj dot roll number right obj dot roll number here yeah obj dot roll number yeah there is a simple mistake is there let us we have corrected it now let us run it.

So, it is now running correctly. So, what we have understood from this demonstration is that in the same file the protector members are accessible to different classes that is all. Now, again if we make this class in different file then definitely it will not work as it works in this case. Now, here is the one example showing the same thing here the class a file is keep in the different file if it is a different file that this will not work for you let us have the second demonstration illustrating the protected member is more there what is the 7.12 program let us load it.

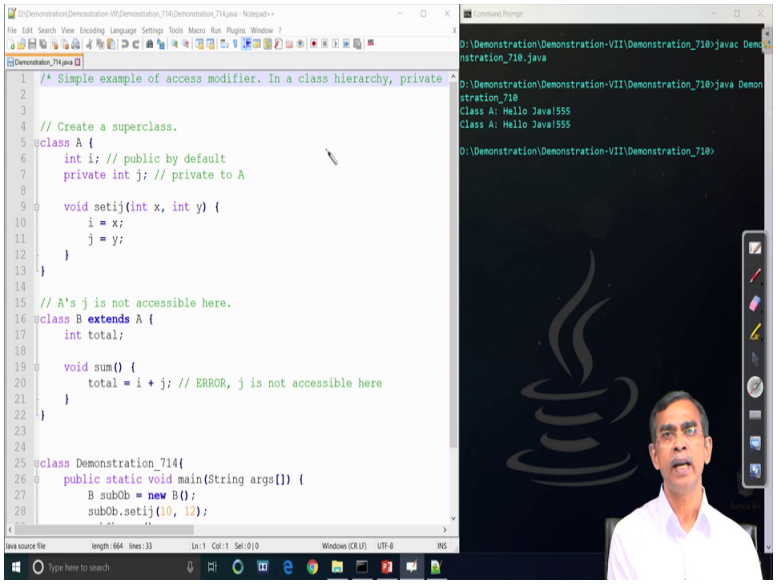
(Refer Slide Time: 26:31)



We have mentioned here protected void display and then yeah. So, this will work for us no issue protected method will be accessible to the same class, but if we store in the

different class if we store the class A in a different class then it will not be there. Now, let us see the inheritance whether we have through the inheritance we can access it or not let us go to the program 7.14 I think this is right we have to skip it go quick.

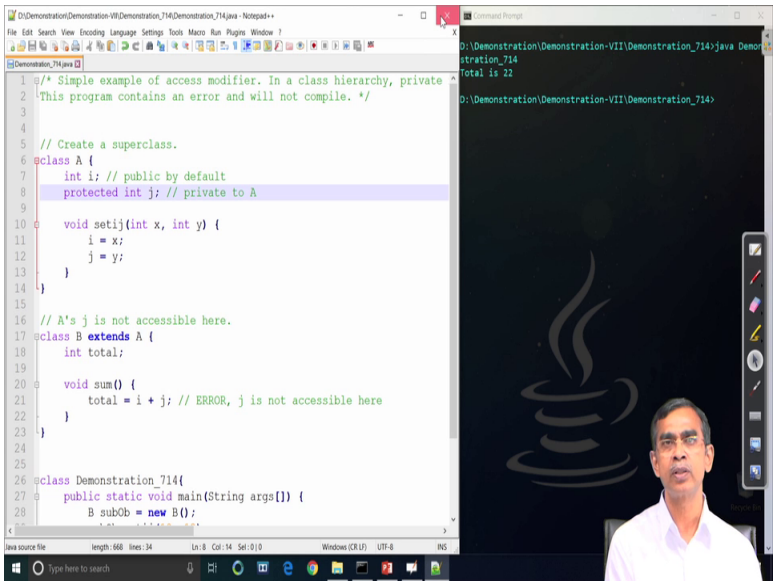
(Refer Slide Time: 27:07)



Now, so, protected member has its more implication in the context of inheritance this is one example; so that how the member protected can be access to the inherited class derived class or subclass. So, class A is a super class in this case where j is a private as is private it cannot be accessed to any other class other than this method in this class. And so, this is a method to initialize the object i and j in discuss. So, it is not an issue because it can access it on the member.

Now, here B class extends A so, here and it has its own member one total as an integer and its method sum now here total i plus j as you understand that j being a private. So, this is and compilation error this leads to a compilation error. So, it will not a work here so, private method cannot be accessible to any derived class right here. So, this will give an error ok.

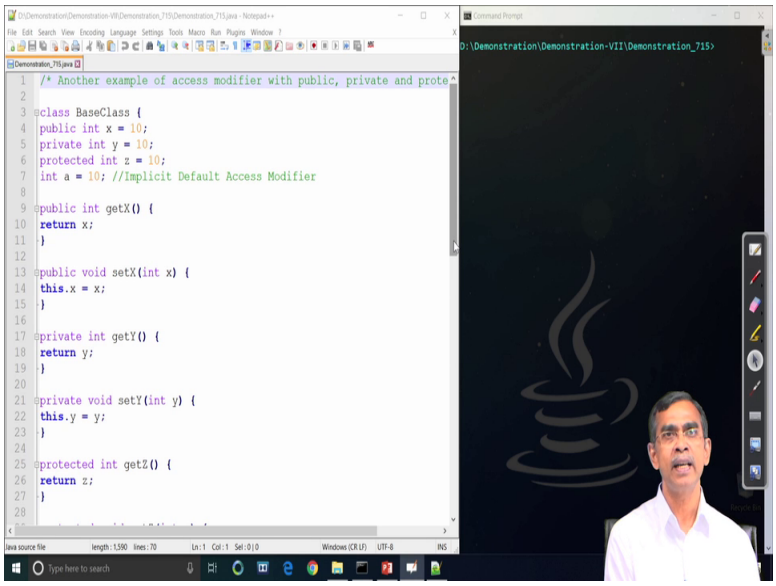
(Refer Slide Time: 28:15)



So, it is giving an error because the j is not accessible there. Now, if we make it protected for example, so, instead of private let us go here protected now see this error can be eliminated here in this case because protected member are readily accessible to the derived class here. So, run this program and now this program will work for you.

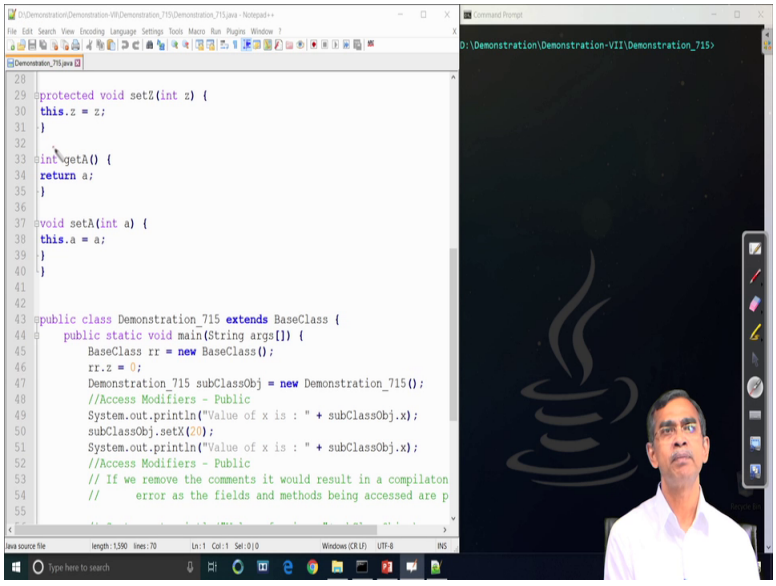
So, this program works correctly and we can see that protected member is accessible to the inherited class. Now, let us have a little bit bigger one example who which includes many other methods with many access modifier 7.15 is a last demo last, but one demo fine let us see the program here.

(Refer Slide Time: 29:15)



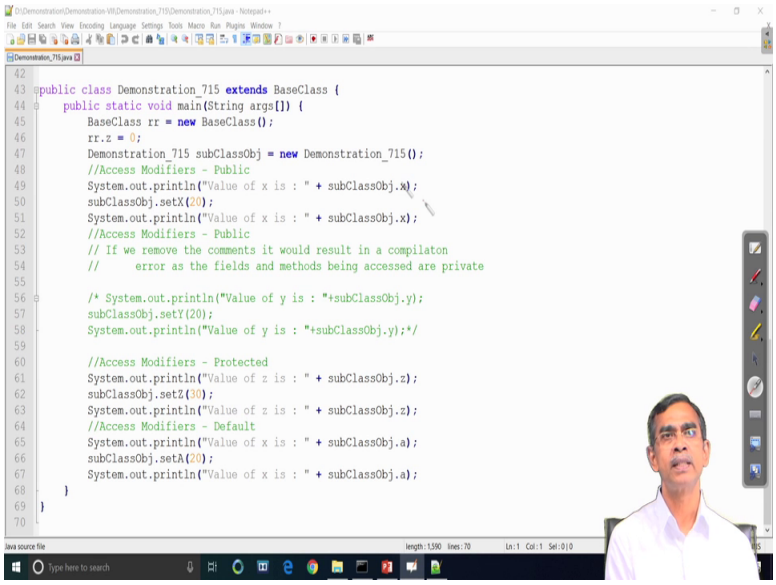
Little bit large program here, but program is readily understandable here. So, we have declare one class called the base class which its members public one, private and protected and integer is a default. Then here is a method public declare as a public getX and then another method setX is also public and another method getY to private it is basically initialized to private value it is a private method and setY is also another private method that basically to initialize the private elements in it. And then getZ is a protected; that means, dealing with the protected members there and then setting the value and then getA return A.

(Refer Slide Time: 29:59)



A is basically the one default values is there seta is to default here. So, we have declared four members public, private, protected and default and then eight different methods regarding they are accessing.

(Refer Slide Time: 30:19)



And here is basically if we remove the comment then those are the comments basically creates a problem. Now, let us have a quick look about the main method in this main method we create an object of the class base class r r and then r r dot z as you know z is a default 1. So, rr dot z is accessible here so, this is not an error so, it will first.

Now, we create a another object of the main class method say demonstration 71 subclass object new this one and for the subclass object we are trying to access the different method get set all these things are there. So, for the access modifier is public is concerned system out dot println it is not an issue it will access because public method public values are accessible to here. So, setX is being a public method it will accessible here similarly getX also will be accessible here. So, this is not an issue.

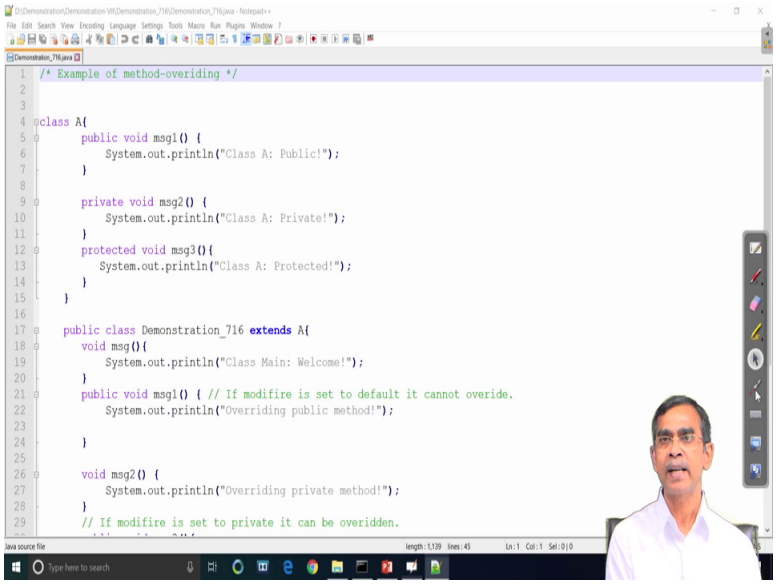
Now, so, far the access modifier public is concerned they are accessible; however, there will be an error if we removed the following comments in the following line right. For example, system dot out dot println value of y is subclass object y y here y is a private as if the private cannot be accessed by any subclass object here. So, that cannot be accessed here. So, if we remove this comment so, it will set an error. So, in that case subclass

object setY also not an privileged a method for this class and then we cannot access it here.

Then so, far the protected is concerned. So, as it is a derived one so, extent because you see demonstration underscore 715 extent base class. So, derived class this means that all the protected member and the protected methods are accessible. So, they all will access here. Now so, far the default access specification is concerned in this class default is in the same file. So, it will not be an error so, it will execute successfully.

So, this is the understanding about the different access modification. So, far the program in Java is concerned now I will just conclude this demonstration with the last example this is related to the method overriding; whether you can override some method having some access specification which is already specified with some other access specifier.

(Refer Slide Time: 32:37)

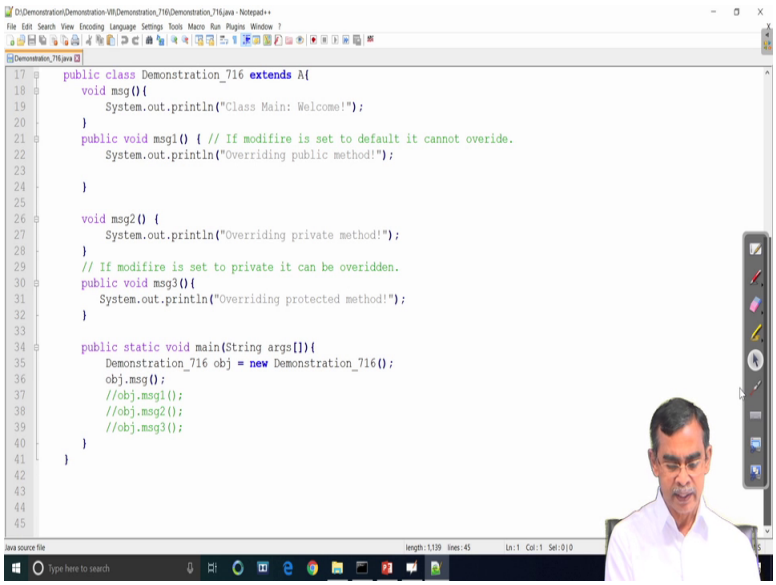


The basic concept is that a method can be override it by one weakest one if it is by if a access specifier is weakest one then the strongest one then it can be specified by the weakest one, but the reverse is not possible. So, this is the concept it is basically upward compatibility it is there.

Now, so, far the different access is concerned private is the strongest and after the private protected and after the protected default and the weakest is the public there. So, if we declare a public later on we can method override as a default, if we declare is a public

overridden as a private can be possible. But if it is a protected and then accessing that is a default it is not possible because, default is weaker then the protected and vice versa similarly public is weakest. So, public default is there then may overriding as a default is possible, but reverse is not possible.

(Refer Slide Time: 33:49)



So, this demonstration has a quick one example that giving the same notation here. Here we declare one class here the class A s class A and in this class we have the method message one as a public msg 2 method as a private and msg 3 is a another method as protected. So, the three different method with three different access specifier we have not used any default access specifier here in this case, but we could do that anyway.

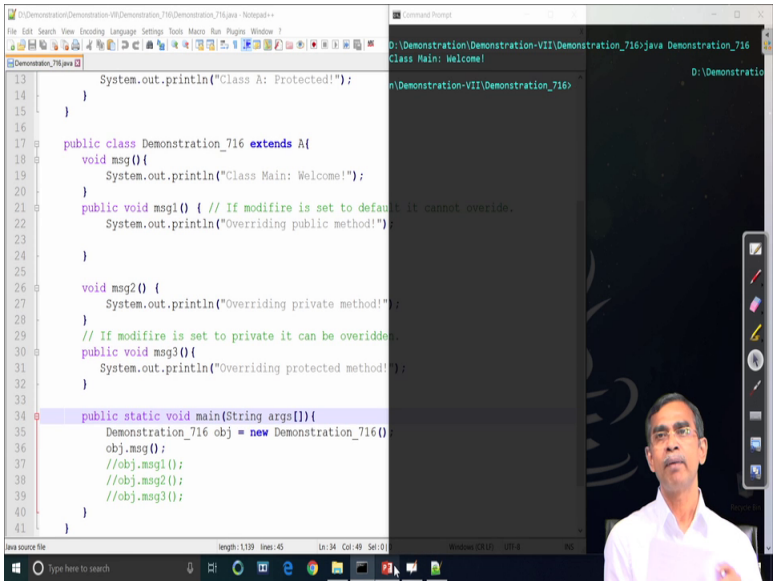
So, let us come to the inherited class extends a now by virtue of inheritance in this class all the method is available except the private method protected is available and then the public is also accessible there. Now, in this method we have declared one method is a default method message it is not an issue. Now, here if you see msg one which is declared in its base class as a private, but here we are declaring as a public.

So, that overriding the method you go to the previous class the base class please message 1 yeah. So, message 1 you see message 1 is declared as a public. So, if it is a public overriding is quite possible public to public same access level method overriding is possible let us go to the main class here main method main class yeah so fine.

So, message 1 overriding is possible, now here msg 2 in base class this method msg 2 is coming as private right go their private method. So, here we are going to a override it as a private we can override it, but other than private we cannot override because it is the highest access phase one it cannot be overridden. Now let us go to the next one yeah. So, msg 3 which is a protected declare as a protected, but here we can override as a what is called the public? So, that is possible.

Now, so, with these things we can override it and then once the overwritten is there we can create the object of these classes and then access it. Now let us compile this and let us see what is the output this program it gives you. So, this basically gives a hierarchy of access overriding method overriding with the different access specifier.

(Refer Slide Time: 36:15)



So, actually for a so, here you can see you know there is a problem you can just see there is a problem in the 716 compile is successfully. Now, what I want to mention as the last notice note is that access modification is not a critical job for there you specify which method which member you want to give a access to for which one. But the compilation itself take care about all this invalid or illegitimate any access specification. For a Java program it is a great relief that if we have wrong things and then program is built up, but this is not a violation because compilation time only it will resolve if we specify according to one.

So, this is the demonstration about the access specifications and we have learned about how the information hiding in access specification is can be done. And; obviously, practice will improve your understanding I advise you to go for rigorous practice for this. So, with this I want to stop it here today.

Thank you very much.