Mobile Computing
Professor Pushpendra Singh
Indraprastha Institute of Information Technology Delhi
Lecture 25
Fragments

Hello, welcome to your next class on fragments. In last two lectures we learned some basics about fragments, in next two lectures we will learn how to create a program for a mobile app using fragments. We will try to implement all the concepts that we have learned, so please look at my screen carefully and try to write your own program as I do this. So number one is that let us first start a program. So I go to the file, I go to new, I go to new project, I give the application name, let us say fragment basics because we are only going to do some basics operations here, I go to next, I choose phone and tablet, then I go next. I create an empty activity. For activity name, I do not really change it and I can just say finish. Now android studio is creating my project.

(Refer Slide Time: 1:43) 2:51

Many of you have mentioned that you are getting problems, which says that such a library does not exist. If all these cases are there, first please try to look at your import statements that what you are importing and second try to look that whether whatever you are importing is actually available or not. For example, you may be importing a library for which you have not added a dependency, so you need to do both in order to make your program run. Now, for example we now have our basic application created, we will make few changes in to it to make it an application for using fragments, currently this is just a basic application which uses empty activity. So the first thing that I am going to change is that, I will change the extends to fragment activity instead of Appcompact activity, so I type fragment activity.

(Refer Slide Time: 3:57)

Now, at this point of time, the android studio gives you a choice to either import the android dot support dot v4 support library or the Android dot App, I will choose the first one because this is the version of support library. Let me choose it and now you see that my import statement has changed and it is saying android.support.v4.app.fragment. Now let me go and add the library to it. If I see the Android studio is still showing red, because it cannot find this library. So now let me go into the file, new, file and then come to project structure, in the project structure I have a small thing called App, so you can see App here.

(Refer Slide Time: 4:34)

So, I go into this, when I click app, I see these tabs, I will go to the last tab called dependencies and in here you can see that appcombat v7 is already there, something which I am already using in the import, but there is nothing like v4 here. So I will press a + and I will choose library dependency. This will open another dialog box and the second entry says support dash v4, com dot android dot support colon support v4 colon 24 dot 2 dot 0, this is the support library that we want to use in our program. I have earlier asked you to use support library called development fragment rather than the android os, there are multiple reasons for this, number1 that the android library support android devices which are as old as running android1.6.

Number2, the support libraries are updated more often. Number3, that because you are using a support library, if a new library comes, if a new version of library comes with an advanced feature, all you need to do is add the newer library in SDK and everything is fine. So there are ample reasons to use the support libraries for development and that is what you should do. So let me select it, press ok, this now appears in my dependencies and I will now press ok and that is it.

(Refer Slide Time: 5:48)

Now you can see that Android has again started building my project and after some time this error will go away as soon as my project is built, so let us wait for it. Yes, so now it has gone. Now, the next step is to look into the layout file and see what kind of layout we currently have. As you see that we have the default layout which Android always gives us. Let me change it to the text, so I would like to change it I would like to make it a very simple layout. I will call it fragment frame layout because in this layout I can put my fragments.

(Refer Slide Time: 6:37)

As we have already discussed that fragments views our inserted into a placeholder and that placeholder is usually provided by the activity, so I will remove this ok I will remove this, I will then add another field Android Id, we have used it at multiple times. So add +Id and I will give it the Id name as fragment container, please remember this name, layout width, layout height I will add. Not going to use any margins, I am not going to use any textview. I am missing one and this completes the layout file for my activity that will be using a fragment. As you can see I am not giving much, I am just giving an Id so that I can refer to this layout in my program. The design is also very simple, it does not do.

(Refer Slide Time: 8:56)

So now we are ready and the next step that we are going to do is to first add a fragment to our program, once we have added a fragment to our program, then we will try to see that how we will bring in that fragment into the activity. So in order to add a fragment into our program, I will go to the Java, I will go to the main package definition, I right click, I go to new, now you will see that I can add a fragment by just clicking on a Java class that is basic Java file and then writing my own code or I can go down to the fragment and I can choose a one of the three fragment options.

(Refer Slide Time: 9:46) 13:07

For the time being we will use a simpler version, because our fragment is very basic. So let me click on Java class, I will give this a name as basic fragment, again I am using fragment as a suffix so that anyone else who looks at my program knows that this class is actually a fragment. I press ok and I see that a basic class has been added. But this is currently a simple java class I want to change it to a fragment, so I will go to first line and I will say it extends fragment, again I have a choice of two, of two inputs of fragments which is coming from support dot v4 and fragment which is coming from android OS, I will choose the v4 version. This will add an import statement and this is all fine.

Now, let me do two things, number 1 if you remember correctly, that we need to first, we need to definitely override a method called onCreateView, so let us override that method. @public void onCreate oh sorry I need to complete this @override public void onCreateView. Now this method takes three parameters, one is a layout inflater, let us call it later. Second is a view group container, let us call it a container and third is a bundle and let us call the bundle saved, instant saved. I am using the default standard names, this is usually a good practice because then you know the variable, what kind of type this variable holds. So bundle again it cannot resolves, I will have to press Alt Enter which will add another import now everything is fine. When we implement the onCreateView, we will have to first create a view and return.

Now, if we try to create a view we will have to call the method which is known as inflate, the inflate method if you see here, let me go back a little. If you see here then the inflate method takes 3 parameters, the first of that is a resource Id, now which resource Id you should provide, well, because we are trying to inflate our fragment, we need to provide the resource Id of our fragment there. But so far we have not created any fragment layout, because we

choose to create our fragment file by just extending a normal java class. So let me create another layout which corresponds to my fragment and whose Id I can give here.

(Refer Slide Time: 13:36) 14:17

So, I will go to layout, I click new, I click a layout resource file and I give the file name as fragment underscore layout or fragment underscore basic to correspond to our name. I choose this, oh I am sorry. Let me first create the layout file, I go to new and I click and XML and under XML I choose layout XML file, in this it ask me layout file name, I call it fragment, fragment underscore basic and my root tag is linear layout which is fine with me, I click it and creates a XML file corresponding to my fragment. Let me show what is happening. So, yes I have created this new file called fragment basic and we will now see what is in the fragment basic.

(Refer Slide Time: 14:37) 15:46

So in fragment basic there is nothing but just an empty linear layout. I am fine with it, I will just add a small text so that we recognize when our fragment has been imported into the activity. So all I will do is, I will just do an add a text and width is fine with match parent, height is fine with let us say wrap content, I will have to add an Id, Android, well actually I need not to add an Id here, because I will not be using it, but let me just say, I want to display a hint which is “enter a sentence” or “enter a word”. So this is a very basic linear layout, let me see its design.

(Refer Slide Time: 16:23) 17:31

It is just linear and a text with a hint of enter a word and let us see also the layout of our application, our activity does not have anything, so when our activity imports fragment, we should see this, ok now let us go back to our basic fragment and now we have something to give when the inflate field asks us, I will give R dot layout dot now you can see I am getting an option of fragment basic. So let me add it, the fragment basic here. The second field is easy, I can just give container here and the third usually is false. When we go further sometimes we will change this variable. A good idea could also be if you want to try why, what happens if you try to convert it to true in some of your examples and see. For view, so now I have added another import statement so now my view is recognized. And then I am doing nothing but just returning my view.

Oh, sorry, I have made a wrong definition, it is not void but it is view. So view onCreateView so now, my basic method is complete. I would like to discuss a very small but important thing.

(Refer Slide Time: 17:38)

Let us go to activity, in activity we have a onCreate method which we override and this onCreate method is protected.

(Refer Slide Time: 17:57)

Now let us go to fragment, in fragment we have an equivalent method called onCreateView and we override that as well but that method is public. Can you understand the reason why? Think about it. As some of you may have guessed, it needs to be public because this will be called from an (())(18:16) so this will be called from outside so this needs to be public. Now we have created a basic activity, we have created a basic fragment; so far we have not imported that fragment into the activity, but let us run our program to see what our program looks now and what it will look once we import the fragment. So, I will use a Nexus xp emulator API23, I keep using it till the nogat android nogat API24 and x7 have been released to the public. So, let me do the run, our emulator is starting, it is taking time now what do you expect to see.

(Refer Slide Time: 19:25) 19:53

Well current example, we have not done much, only thing that we expect to see is the basic layout of the activity. However it is important to see that, so that when our activity actually brings a fragment then we can identify the reference, so let us wait, so yes, our activity has started and there is nothing here, as expected.

(Refer Slide Time: 20:08) 21:36

Now, let us write rest of the code to bring in the fragment into the activity. Earlier I had discussed that there are two ways to do this, one is to make changes in the XMl file of the activity, number 2 is to write a code which brings the fragment and we also discussed that what is the positive open approach and what is another and normally if you want to work with the fragments dynamically it is always good to use code and that gives you the flexibility. Now, let us start writing the code that will bring in the fragment to the activity. First I need fragment manager let me call it fm or let me call it fragment manager. I get the fragment manager by making call to a method called get support fragment manager.

Second I need to create the fragment object, Create the fragment object and the first thing I try to do is, to find that if there is such fragment already in the system. Let me just call fragmentmanager.findFragmentbyId and we will pass it the Id of the container that it has.

(Refer Slide Time: 21:45)

So if you remember I had asked you to remember this Id, this refers to a layout of the activity. We will pass it because we want to know if there is any fragment with this Id. So R dot layout dot R dot sorry Id dot fragmentcontainer.

(Refer Slide Time: 22:26) 24:44

And once we do this, we need to make a check if it is returning null or if it is returning a non null value. Because your activity goes into different life cycle stages, so when you start your program it will obviously be null the very first time. However, once you will start interacting on your Android device and your activity goes into different life cycle stages such as start, stop, destroy, etc, there is chance that in one of these activities, one of these states, your activity is hidden but it is still alive, in those cases this value will not be null, so that is why we make this check. So if it is null, we create a new fragment and the let us call it new basic fragment.

And now I want to start a transaction which will add this fragment to the activity. Fragment manager, we will begin transaction and I want to add a fragment, now I first need to give the Id where I want to add the fragment and then the fragment that I want to add. So the first parameter means where I want to add and second is what I want to add and after that I do commit and that is it. Hopefully everything is fine. Let me run the program. Ah ok, so I made a mistake, I done extra curly bracket, so now everything is fine. Let me compile and run our program.

(Refer Slide Time: 25:02)

So our program runs and here you see, now our views changed, our activity has actually brought in a fragment. So this is not just the activity layout, this is the fragment layout that you see. So this is the basic fragment application which is showing you how to bring a fragment in an activity using the code. Now let us add some log messages to understand the different state transitions between activity and fragment and between the different states of the activity (())(25:34). Just like last time we are going to add log statements and see how

they are played. Let us first stop, we go to our main activity and we try to extend all the other methods, we try to override them.

(Refer Slide Time: 26:07) 27:46

First I will also have to declare a private static final string tag = main activity and then we have added the same methods, I have copied them from my profile. Now we go to the fragment, we first add the tag which is again the same, private static final string tag = basic fragment. We now override all the other methods; so, many of the methods are common with the activity, but then there are methods which are different. For example, there is onCreateView and there is also an onCreate so, let me create a onCreate method. So unlike the, just like the activity onCreate takes a bundle saved instance state.

(Refer Slide Time: 28:55)

Not going to do anything, except call in the onCreate, inside onCreate. So we have onCreate, we have onCreateView, we have, now there are some other methods as well such as on Attach and onActivityCreated in the fragment, let us also do them. So, our first method, on, so onActivityCreated, this also takes a bundle, let us just call it b then just call super dot onActivityCreated bundle inside onActivityCreated. Another newer method for this method which exists in fragment was not in activity is onAttach, takes a context, let us call it c, it also asks me to import another file. It should call, just like other methods, superonAttach onText and I will change the log and entry to Inside onAttach.

(Refer Slide Time: 31:00) 31:44

Now let us see if we are missing something, so actually we are, there are 2 more methods, onDestroyView and onDetatch. So, yes, we have onDestroyView, we have onDestroy and we have another method called onDestroyView, so let us implement that onDestryView view, ok, what it is showing. Ah, I typed it twice and then the only method left is onDetatch, so let me also change that to onDetach as well onDetach ok, onDetach and inside detach. Fine, so now we have (overrid) overridden all the methods of the lifecycle of the fragment. We have also overridden all the methods of the lifecycle of activity and we have added log messages to them.

This is not necessary in a normal program to override all the methods, but we want to see the interplay between the fragment and the activity; we also want to see how and when different lifecycle methods of activity and fragments are called. So let us clear our log and run the

program now. We will now concentrate on our log window because we have already seen what our program displays. We will just concentrate on our log window to see that how different methods and when these different methods are called. Let me start the execution and let us start watching the log window. So I will set set a filter of Inside, and I am hoping that, ahh, ok, so there was some log but we did not see it, I want to see why it happened. Oh, so I think we, by mistake we removed all our log, let us stop, start the execution again and see.

(Refer Slide Time: 33:52)

So, now we start, yes, now we can see. let us try to put a filter of inside, so now you see that the very first, our fragments onAttach, onCreate, onActivitycreated and onStart gets called before the onStart and onResume of main activity is called, but after the onResume of activity is called our fragment onResume has been called. This is as we learnt last time in our lecture let me open those slides for you as well so that you can once again look at it.

(Refer Slide Time: 35:27)

So, as you see, this is what is happening, we knew that, let me setup correctly so yes you see for the fragment on Attach, onCreate, onActivityCreated have been called with the created aspect, then the activity started, activity started on Start is called of the fragment, Activity resume, onResume is called. Now, let us try to interact with our application. What happens say when we press home? When we pressed home, you can see that out activity, our fragment went into onPause and our activity went into onPause, our fragment went into onStop our activity called. This is again as we discussed here, when the activity is paused onPause, when the activity is stopped onStop.

(Refer Slide Time: 36:47)

Now, let us bring back and again you see start onResume , then let us kill it, now you see that onPause, onStop as you know that onPause and onStop they are called before the onDestroy and the activity before the activity call goes into onDestroy, fragment has to destroy its view going to onDestroy and Detach and then the onDestroy of the activity is called, so something like this onDestroy, onDestroyView so,onDestroyView, onDestroyView, onDestroy, exactly as we learned last time.

So, now you have created a very basic fragment and you have called it from an activity, in the next lecture we will make it slightly more interesting where we will create 2 fragments inside a same activity and then we will also call activity from fragment and we will also call a

fragment from a activity. That is we will use a callback method to take the transfer from the fragment to the activity and similarly from the activity we will get a reference to the fragment and come back into the activity. Thank you and keep going.