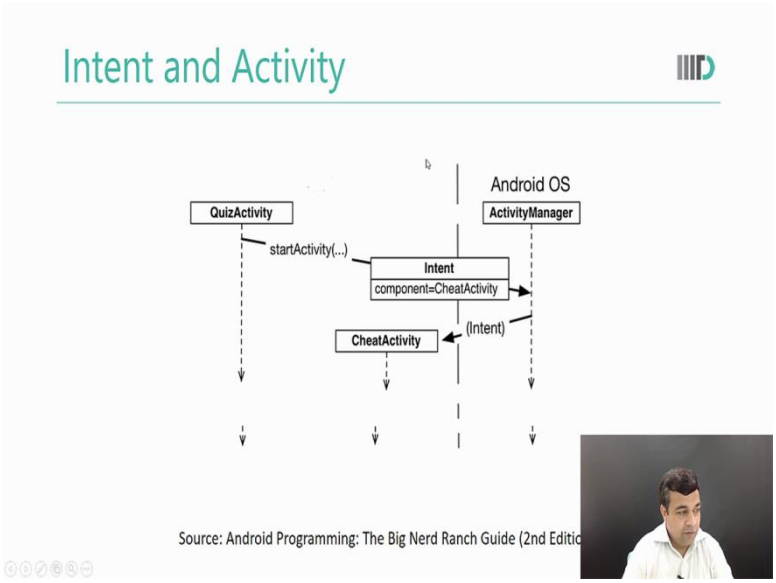
Mobile Computing
Professor Pushpendra Singh
Indraprastha Institute of Information Technology Delhi
Lecture 20
Intents

Welcome to your class, so far we have learnt how to create android applications which consists only a one activity. However, advanced applications are rarely made of just one activity and most of them applications have more than one activity. Today we will learn how to create multiple activities in a single application and we will also learn how one activity talks with another activity, how one activity starts another activity, and how can activities pass data from one activity to another both when starting an activity and also when an activity finishes how can it return back the result. Let us see through a simple program and a small presentation.

(Refer Slide Time: 1:06)

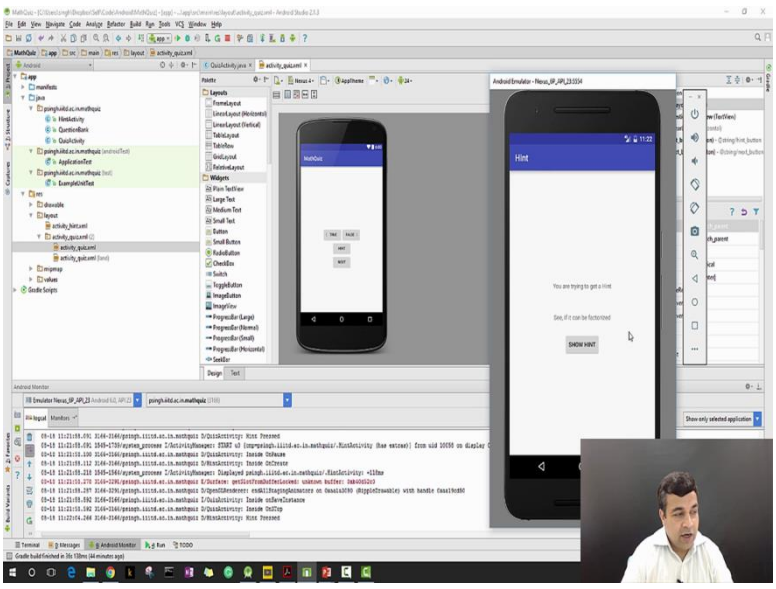
The way 2 activity talk to each other is through what we call Intent. An Intent is a message object that a component can use to communicate with the OS. If you remember from our discussion on components, android applications usually have one of the 4 components Activity, Background receivers, Services or Component providers. Intent is the object that this component can pass to each other in order to communicate. Now, Intents are a sort of a multi-purpose communication tools. The Intent class provides different constructors which we may use as we need them.

(Refer Slide Time: 1:54)



So, how does Intent work with activity? Well, as we will quickly see, suppose this is your activity which you originally have and you want to start another activity. From your activity you will use a method called start activity and you will pass Intent into it. This will go to the android OS activity manager which will then pass this Intent to another activity and invoke that event. Let us go into our program and see how it all works. I would like you to pay extra attention because today's topic is different than what we have been discussing earlier and it is a very important topic, let us first see our program that we had made.

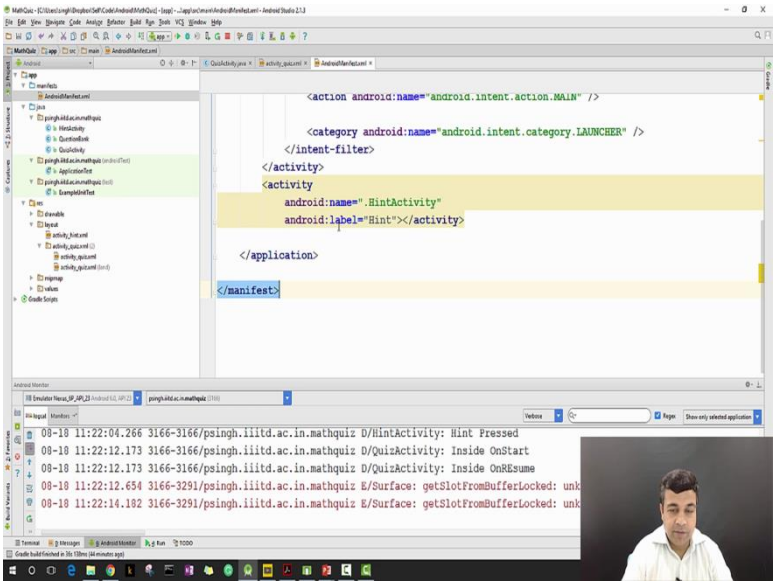
(Refer Slide Time: 2:47)



So, this is our math quiz program that we earlier made. I have changed the math quiz program by adding an extra button here as you can see. And what this extra button does is

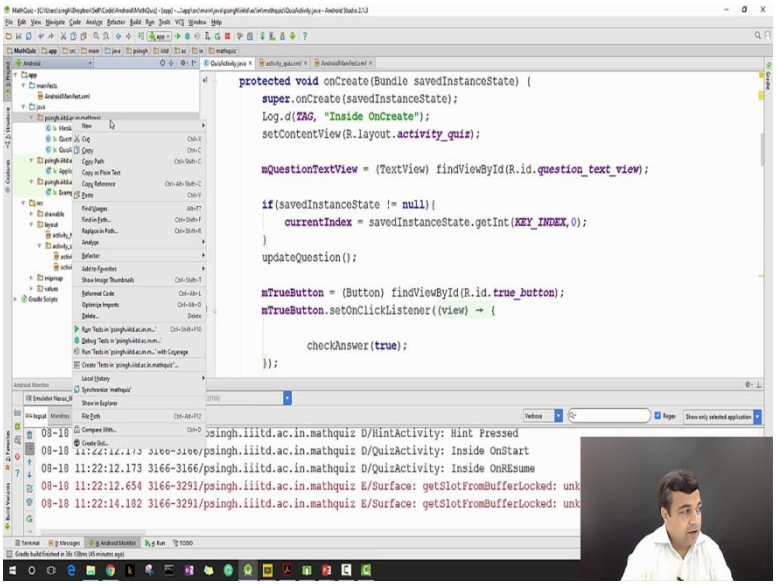
that is that it starts another activity. So, let us first see an execution of our program, as you can see the program, the rest of the program is same. However now I can press this extra button, it takes me to another activity, I can do some interaction on another activity this is another activity it is a hint activity and I can come back to my original activity. So, now our math quiz application that we have been working so far has two activities instead of one.

(Refer Slide Time: 3:54)



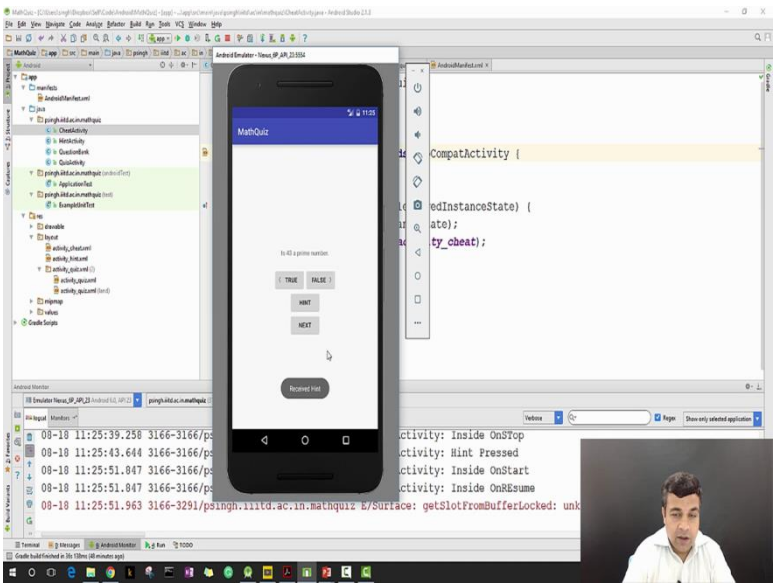
We can further check it by checking the manifest file so if you look at our manifest file, earlier we only had a quiz activity now you see that we also have another activity. So, our application has 2 activities and we are going from one activity to another, coming back to the previous one and we are also passing it. In today's lecture we are going to see, how to create a new activity, how to communicate with the activity, how to pass information to the activity and how to get result back from them so let us start. We will start by adding another activity to this program which we will call as Cheat activity. So let us start with the quiz activity class this was our original class, I want to add another activity.

(Refer Slide Time: 4:43)



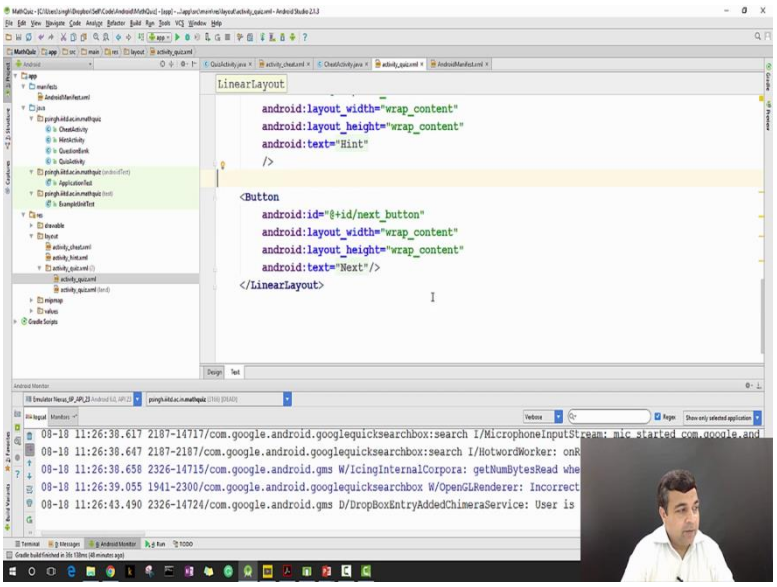
I will go to the beginning of this folder, right click go to new, inside new I will see a menu item called activity, in that menu item I will choose empty activity. Now, a dialog box has opened and it is asking me for some values, let me give the activity name as cheat activity. I am not making it a launcher activity. My layout name is changed automatically and I press finish. After I have done it android studio has added another activity to my program. Where do you think our project has changed? Try to think about it a little bit. Ok so whenever we add an activity we are adding a new Java class which in this case is called Cheat activity. There will be a layout XML file corresponding to this Cheat activity and if we check our manifest file we will also see another entry in the activity list for this new activity.

(Refer Slide Time: 6:29)



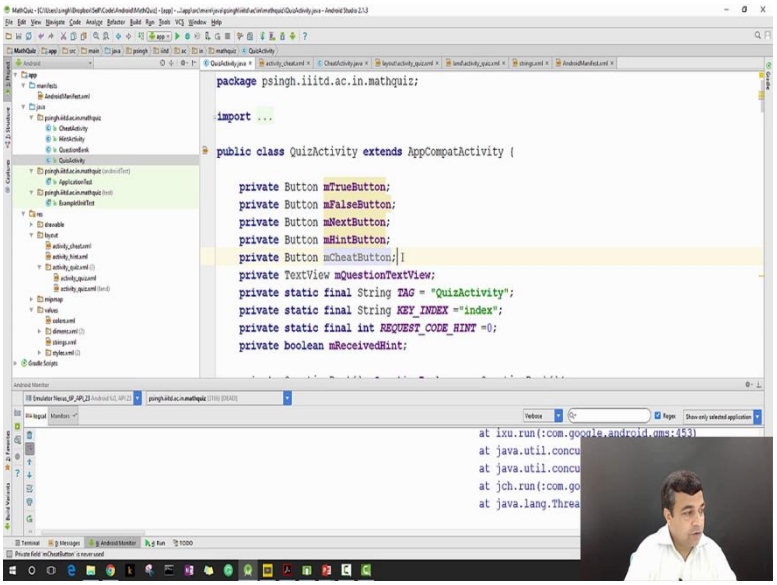
So, over all our project has changed visibly in 3 places. A new Java file, a new xml file and a new entry in the manifest file, now Let us start adding some functionality. So, what do we want to do with the Cheat activity? Let us first see that what all we doing with the hint activity. When I press the hint button it takes me to a new screen which is another activity hint activity. When I press a button there it gives me a hint to find out whether number is a prime number or not and then I can come back. Now, with my Cheat activity I would actually like to add a button here just call cheat, so when I press cheat I want to go to a new screen, I want to get an initial message that I am now cheating in my exam but I still want to continue, I should get the direct answer that whether the getting number is a prime number or not. And when I come back my main activity should know that I have gone and known the answer using cheating and I am not answering there directly.

(Refer Slide Time: 7:46)



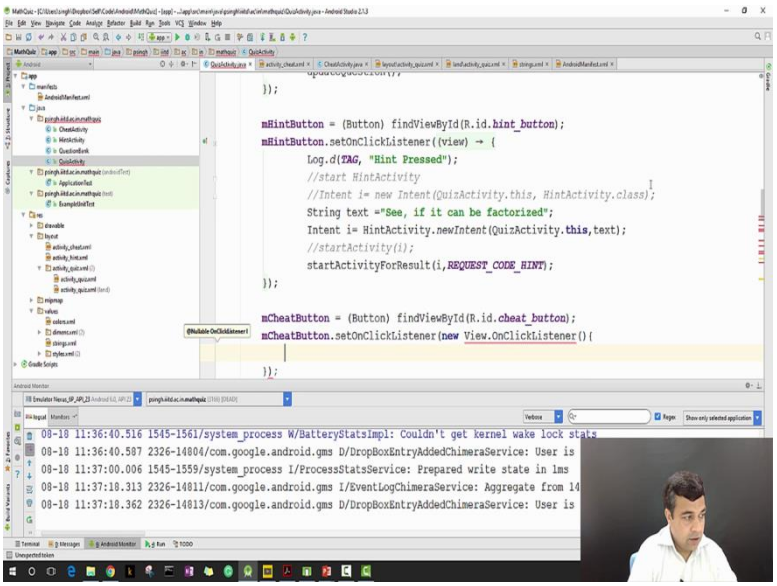
So, this is our aim and this is the program that we are going to develop now. So, our first step in this is to add a cheat button in our quiz activity class. This was our quiz activity, this was our xml of quiz activity I want to just add a button here. So let me just add on the button here, I will ok yes Ok so let us type a new button definition here. I am adding a button, calling it cheat button. I am giving an id to this button because I want to impact with this button later on. Layout, width and height are just enough to help the content. Now, I would like to display the text – cheat button. I have not yet described this text let me go and yes, we have got a cheat button and I am seeing cheat here.

(Refer Slide Time: 9:54)



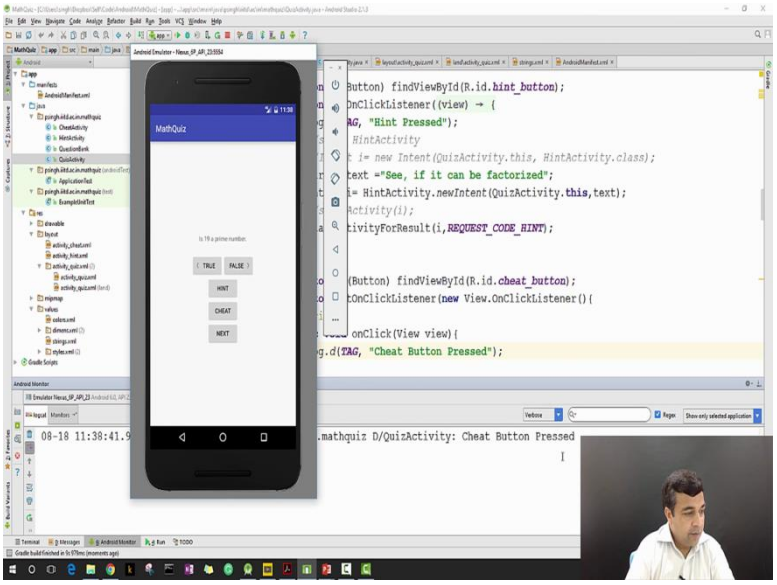
Let us just quickly look at our XML looks but this looks so light. I would just like to see how would it look if I do a landscape and it looks fine. I have hint, I have cheat, I have next that is ok with me. So, now we have got our cheat button let us first quickly check whether it is working or not. Let me go to my quiz activity again. In order to make use of our button we will have to describe a variable called cheat button, I am describing here, private button and cheat button. Then we will have to go and we will have to add a cheat button on click list. So we have done that previously let us just do it again, mcheat button = button find view by id R.id.cheat_button.

(Refer Slide Time: 10:57)



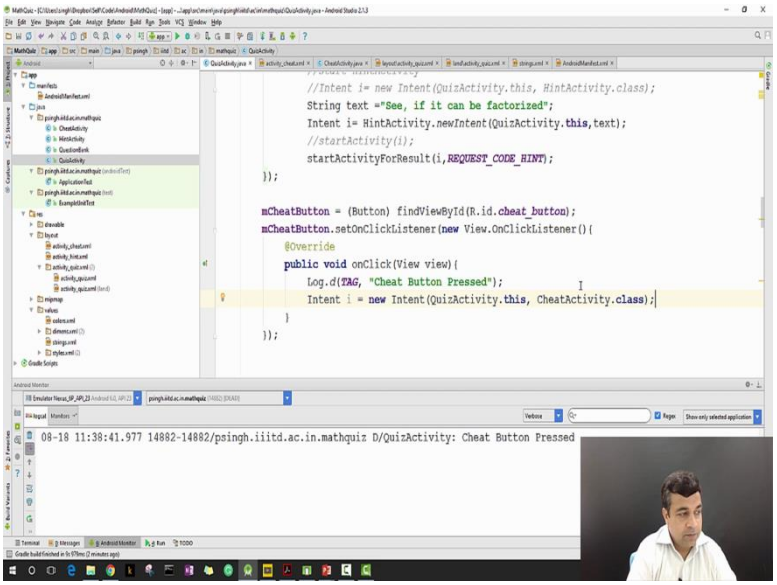
Now I want to add a listener to it mcheat button.setOnClickListener new view.onClickListener, I would like to override public void onClick method, I would just add a log message here log.d TAG cheat button pressed, now let me run our application again and see if we get the log message.

(Refer Slide Time: 12:12)



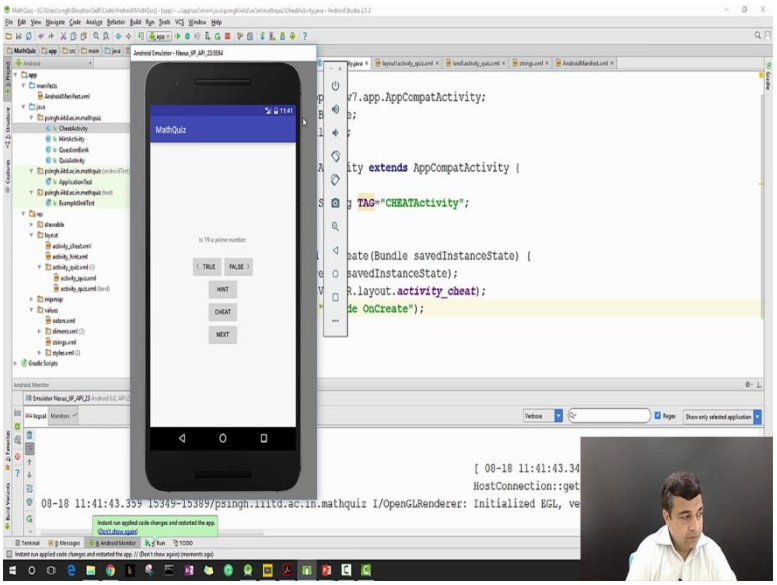
Ok let me first clear the log tag, when I press the cheat button I am getting my log message so, everything is working fine. Now, what I want to do is I want to start my Cheat activity when I press the cheat button. So, if you want to start one activity from another you will have to first create an Intent object and then you will have to call a method called start activity which will take that Intent object as one of parameter and then your new activity will get started let us see.

(Refer Slide Time: 13:25)



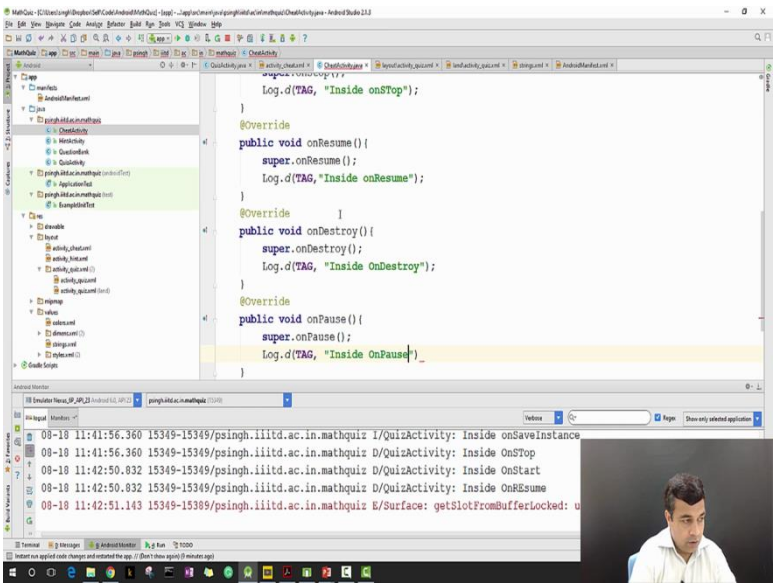
I do very simply Intent i = new Intent, I am creating an Intent, my intent function takes two parameters, one is a context parameter. In this case my context is my present context which is the quiz activity and the class of the new activity that I want to look. As you see, new intent has two parameters, I have given both of them here, only thing that I now need to do now is call start activity that is it. Now let us go to our Cheat activity and we should see if our Cheat activity starts or not when we press the cheat button. In order to check that I would like to add some log messages so, first letter I create as TAG, then I just do a log TAG Inside onCreate. You can see that the log has been turned into a red; it means I need to input correct libraries in android studio I will press alt enter and the android will import it automatically. Now let us run our program again and check whether our activity gets lost or not.

(Refer Slide Time: 15:13)



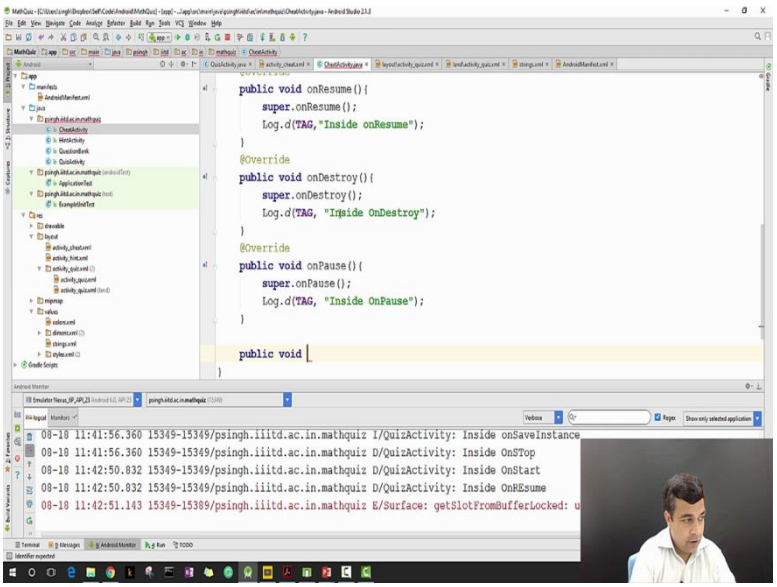
This is our original application, let me clear the logged out window. I pressed cheat and if you see the initial 3 lines. In quiz activity it says cheat button pressed, then in quiz activity it says inside onPause which means that my quiz activity is now in a `Paused' state. Then it says Cheat activity inside onCreate. My Cheat activity has been started. Later on my quiz activity will move from the `Paused' to the Stopped state because it becomes completely invisible, but my Cheat activity has now been started. And as you see that my screen is now changed to my Cheat activity, let me press back button here. And I have come back to my quiz activity which went from start and resume. So, we started one activity from another and came back to previous activity.

(Refer Slide Time: 16:48)



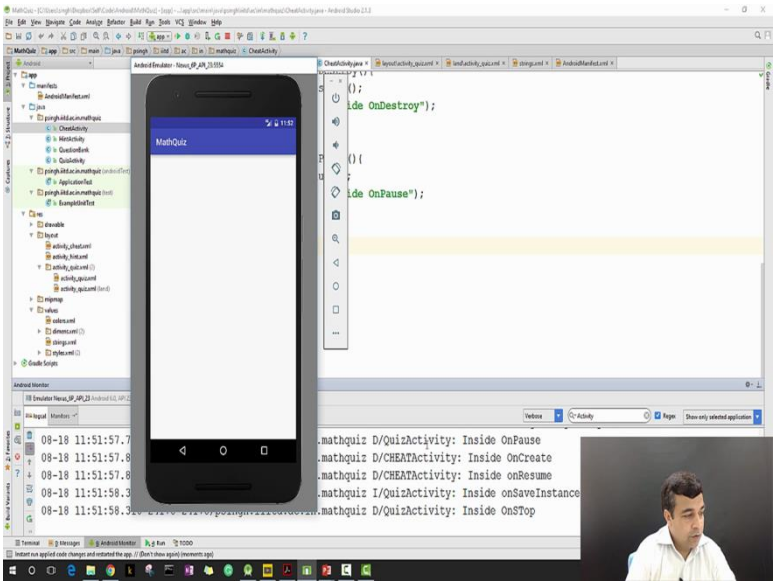
Uh in order to understand the application life cycle better or the activity life cycle let us also over write the methods what we did last time and put appropriate log messages to see how activities behave when they move from, when the program move from one activity to another, so let us extend the methods that we did earlier for quiz activity. Let me do add override public, public voidonStop super.onStop now I just add a log message here side stop. Public, void onResume super.onResume log.d TAG Inside onResume @ override public void onDestroy, super.onDestroy, log.d TAG inside onDestroy @ override public void onPause and onStop, on Pause super.onPause, log.d.TAG inside onPause.

(Refer Slide Time: 20:15)



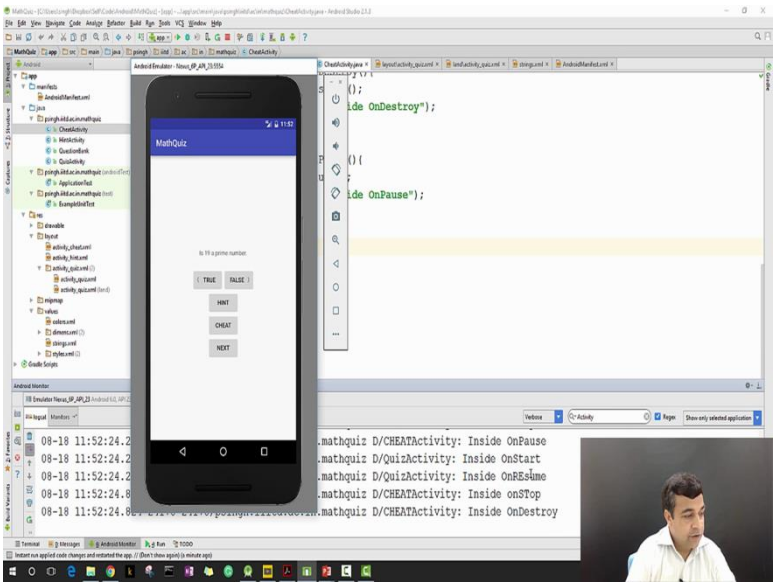
Public void onStop, super.onStoplog.d TAG inside onStop. Why it is giving error? It says it is already defined it, so I have already defined. So that is fine, now let us run our program again, and this time tries to follow the path of the activity life cycle of both of our activities.

(Refer Slide Time: 21:34)



So, we start, go hopefully ok so, it is launched very quickly, and let us clear the log tag. Just said it for activity as a filter, press the cheat button, moment I press the cheat button I can see that from quiz activity went into the Pause mode Cheat activity started, Cheat activity is now in Resumed mode and my quiz activity is now in Stopped mode. What happens when I press the back?

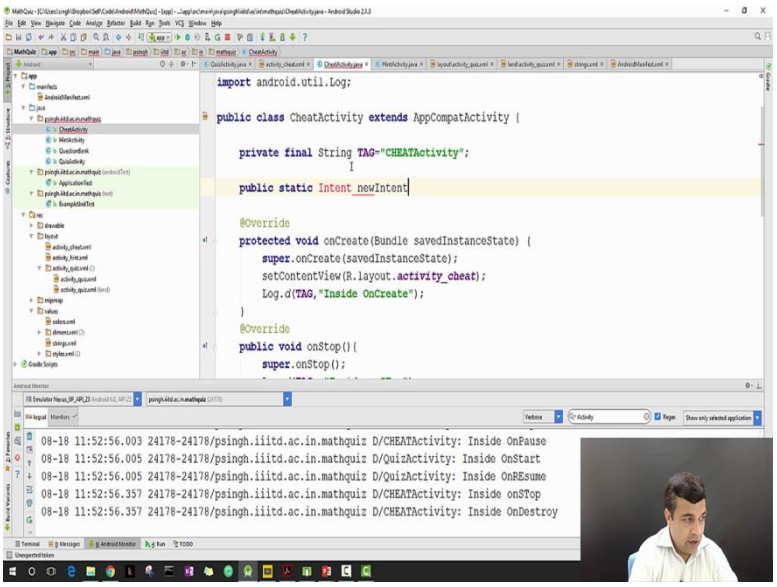
(Refer Slide Time: 22:06)



When I press the back button my Cheat activity goes from pause to stop to Destroy and my quiz activity from stop goes to start and resume. Let us do it again, Cheat button pressed quiz activity Paused and stopped, Cheat activity created and resumed back button cheat activity paused and destroyed, Paused, stopped and destroyed while quiz activity started and resume

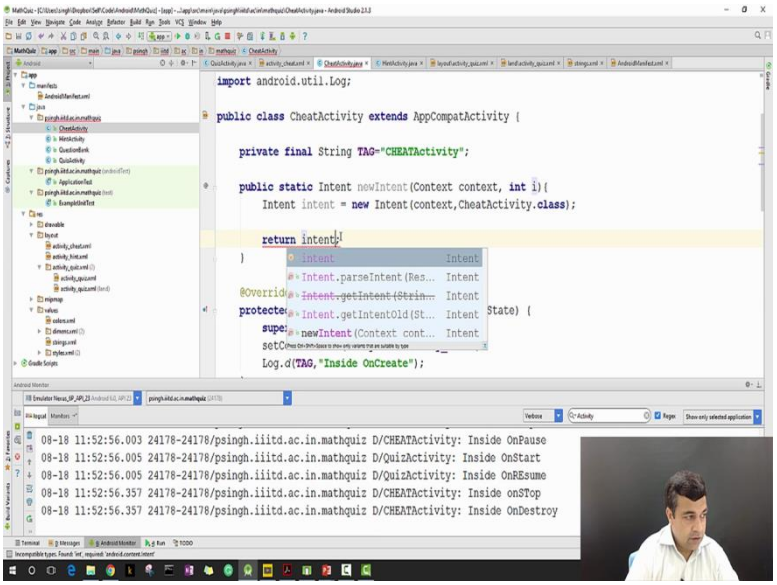
again. So this is about the activity life cycle when you go from one activity to another. Now, let us do some more work. So far we have seen how to go from one activity to another and come back. Now, we will see how to pass data from one activity to another and then we will see how to get result back from one activity to another. So, when we want to pass a data in to another activity we use a slightly different technique. Number 1 that we will try to create our Intent object such that we can pass the data and then we will start our activity.

(Refer Slide Time: 24:11)



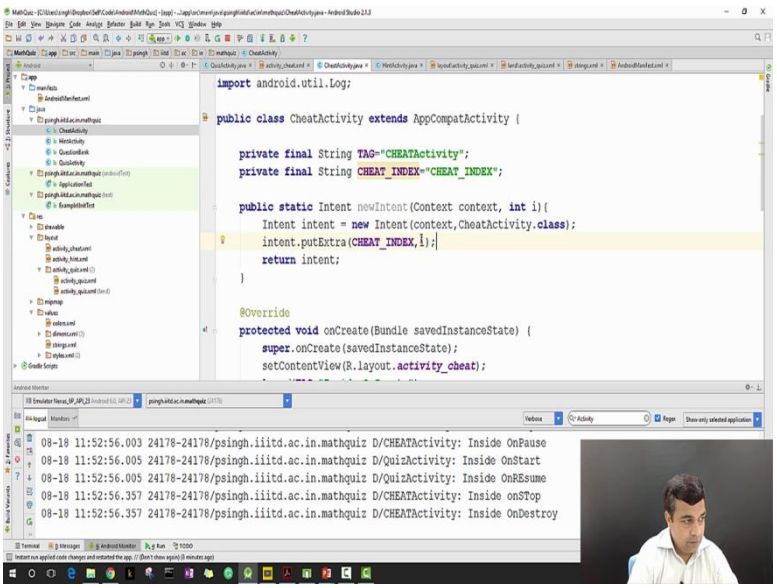
So, we will stop our original Intent object. Now let us go to the Cheat activity and create a method a static method which will help us to create a new Intent object with data that we want to pass to our new activity. We will call this method as new intent method, we will start with public has to be static must written an Intent object and let us say its name is newIntent. As a parameter it should take the context that we were trying to pass earlier, so let me pass the context here which is context then I have a choice to pass the value that I want to pass from my activity to the joint.

(Refer Slide Time: 26:19)



Let us say that we want to pause currently a number. This number could be the number of our index of the area that is displaying the questions. So, what I will do is I will type as int I, it is giving me two warnings because I have not imported correct libraries, I will import both now it is fine. Inside it I will have to create an Intent object and return, so let us do that. Let us create an Intent object i which is equal to new intent context , cheat activity.class, written I Oh and we are defining integers here so, I will change into intent.

(Refer Slide Time: 28:09)

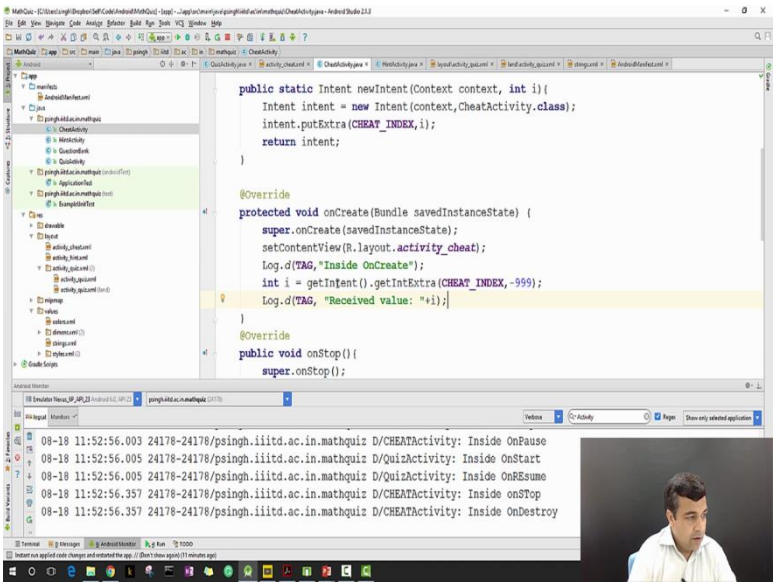


Now if you want to pass some value to another activity we use what we call as intent extras, let us just see how to use intent extras. What I will do is I will do intent dot, If you can see there is a method called putExtra, let me call up a putExtra method, and puExtra method

allows me to put a key value pair. I want to choose the key value pair with the value is of type int, let me use that. For that Number 1 we will have to also define our key so, I will just go and define our key which is let us call cheat key = the value cheat_index, I would have some name it as cheat index, I will pass it cheat_index,i.

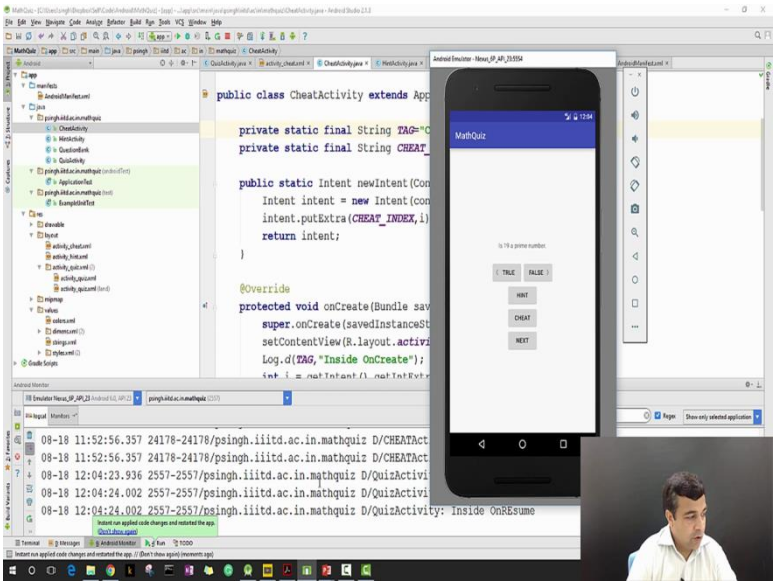
Now, when this new intent object is created, it will also have this value i. I want to pass this value i of this integer value from the parent activity. So, let us go back to our parent activity class which was our quiz activity class and instead of calling the earlier new Intent method, I will now call another method just intent i = cheatactivity.newIntent, I will passes quiz activity now I can pass them the integer value. Because I want to cheat on my exam I would like to pass it the integer value of the index of the area that is displaying the question. So, for me this value is stored in the currentIndex.

(Refer Slide Time: 31:35)



Here I will start, let us say activity ok. Now we have paused this value, we have to still figure out how to make use of this value. So, in order to make use of this value, while I do is getIntent method, getIntent method returns me the Intent object that was passed when this activity was invoked and corresponding to the putExtra method I get methods which are called getExtra. So, I will be using a method called getIntextra, getIntextra method will take the key by which the data was stored and will take a default value. Let me take the key as the key which I created earlier. Let me said the default value = - 999. And let me store it into another int. I will do nothing much but I will use another log message, now I say TAG and I will just say Received value + i. Now let us try to run our application again and see how it works.

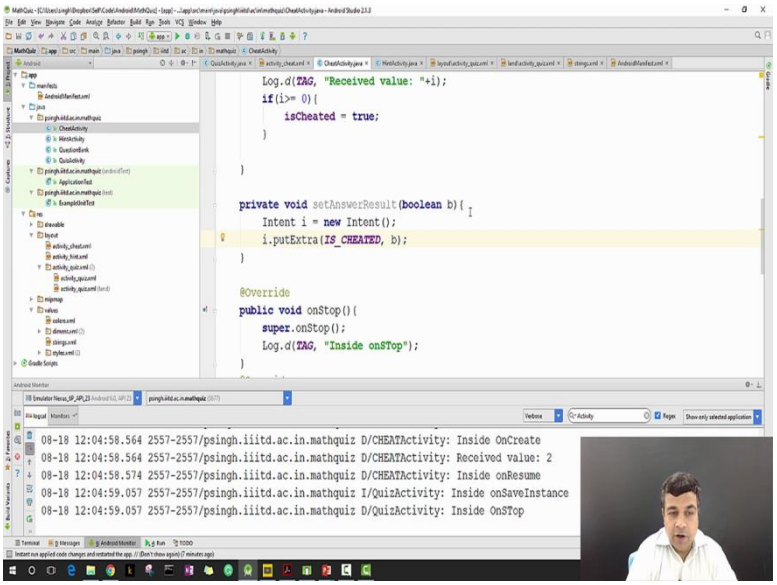
(Refer Slide Time: 32:36)



So, we start Oh! So, we are getting an error, let us check what the error is. Wow! Ok. So, I have described it at final but I have not described it as static. I must describe it as a static, so that I can use it in a static function. Let us hope that now we will not get this error in fact this is printed also here as static, let us run our program again and as you see this time, our program has launched, I will clear the log, everything is ready. We want to now press the cheat button and we want to see that whether we are seeing the value or not. Press the cheat button it says Received value = 0 because we know that we start with our index 0. Let us go to some other question, I have now come to third question, now let me press cheat button and as you see now I am receiving the value 2. So I have been successfully able to pass the value of my current index from my parent activity to my child activity.

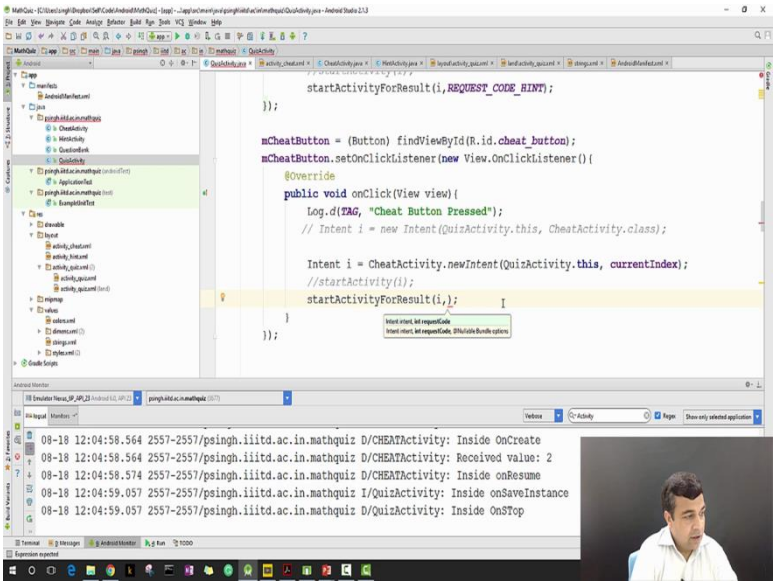
Now, we will move to the second stage where we will pass something back from the child activity to the parent activity. For that we will have to make some changes in our cheat activity class. Our cheat activity class will actually have to use another Intent object, put some value into that Intent object and then that Intent object will be used to pass the values to the parent activity class. Ok Number 1 let us assume that for the time being we just want to return to the parent activity that either the user pressed the cheat button or not. What is the way to check it? You would like to check it by using this i value.

(Refer Slide Time: 34:38)



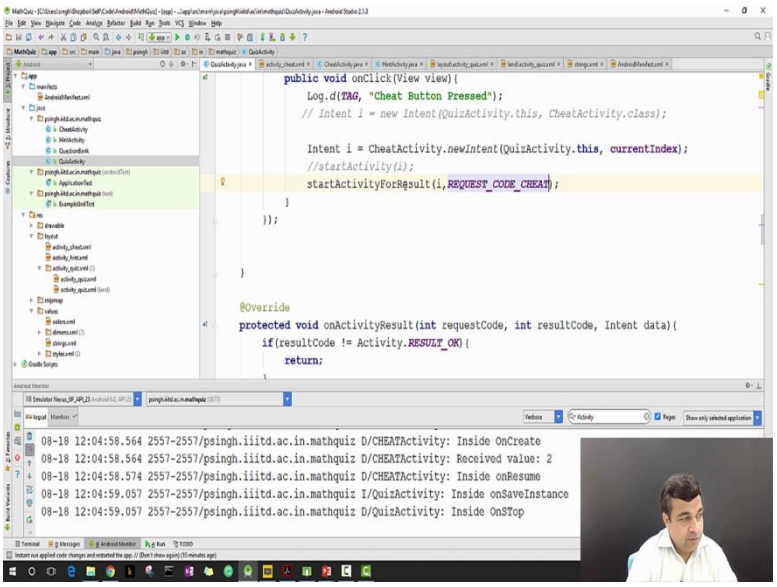
So, let me just say that if i > = 0 then set a Boolean variable. I will define a Boolean here, I write Boolean is cheated = false at if i is 0. Now I will say is cheated = true and after that, I will call a local method. Let me call my local method, set answer wizard and my local method does nothing but it will create uh once a result it will create an answer result which can then be use to send data back to the parent intent. This takes of type Boolean b, now I need to create an Intent object, again I am creating intent, new intent but the next step is similar to above where we have put an extra value here, we will put an Extra value here I.putExtra again this will require a key value pair and you will have to create a key, let us go above and create a key private static final string is cheated = is cheated. I will come back and say is cheated and I am using a Boolean b that is it.

(Refer Slide Time: 37:39)



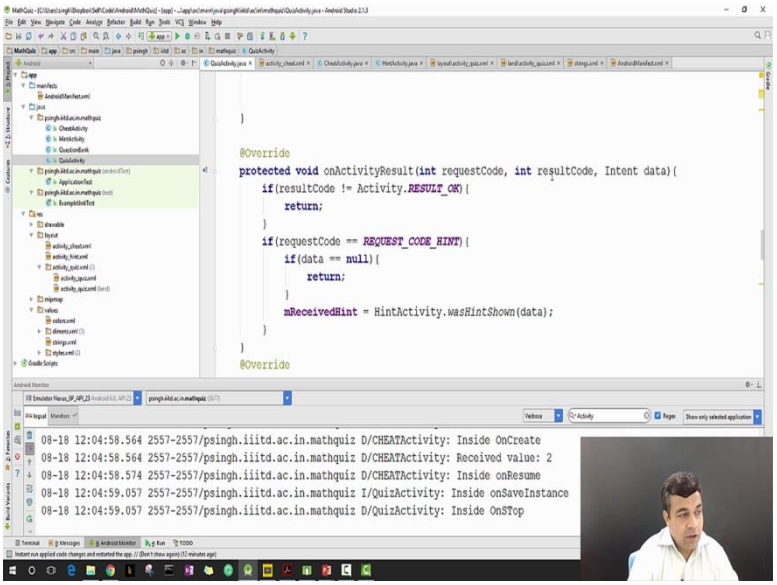
Now, I have put my value that I want to pass the parent activity in to an Intent object. All I need now is to call a method called setResult so, set it result method takes 2 parameters. Number 1, it takes a integer code there are 3 integer code defined, one is a result ok, one is the result cancelled another is a result first user. I will explain them later for the time being we will take the result ok and the second parameter is actually an Intent object, so that is our i. Now whatever value we wanted to return we have put it and we have called setResult. In order to receive data from setResult, actually our quiz activity should have started activity in such a manner that it can receive the data.

(Refer Slide Time: 39:10)



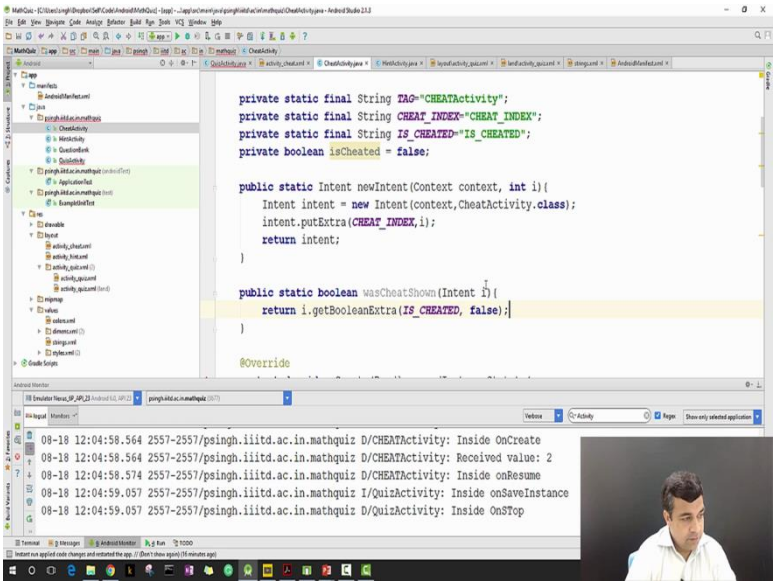
So, we will not be using this method anymore but we will be using another method which is called startActivityforResult and I will call it i, which is my intent, then I need to provide a request code. The request code is nothing but just a unique integer, so that I know that which of my activity is returning me the result. So, this again has to be a constant integer, let me first define it. I will go on top and I will define a private static final int request code from my Cheat activity, now I will give it a value. Let me come back I will give this request code cheat here. Ideally now, I should be able to receive the value.

(Refer Slide Time: 40:48)



So, what do we do when we receive this value? Well, we can do few things. Well, first thing we have to do is to handle what we want to do with the received value. We are receiving a Boolean value so let us first find a way to store that Boolean value. We have to find another private Boolean variable which is mcheated and come and now the second step that I need to do is to override a method called onActivity result. onActivity Result takes 3 parameters. One is a request code another is a result code and another is an intent data. The result code corresponds to the result that you are returning in the child activity. For example, we returned the result as the result ok. The request code corresponds to the request code that you have defined in the parent activity. We have defined 2 request codes for our cheat class, this is our request code. And then the intent data is the data that you will receive from which you will retrieve the values.

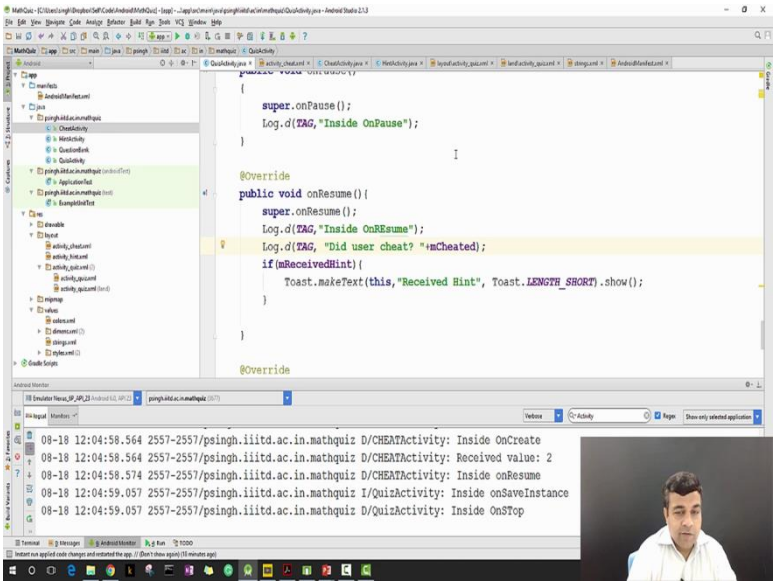
(Refer Slide Time: 41:52)



So, I have already over written it, let me now add code which will get invoked when we receive the data from our Cheat activity. Again if data = null Return otherwise set the Boolean = whatever value you want to receive from the data. So, let us see how to handle this one. Let us now go back to our Cheat activity class, let us go back to our Cheat activity class and add another static method Boolean. Let us say wascheatshown if you take an intent i and just like we had used the putExtra, we will use the returnExtra. getbooleanExtra the boolean was cheated and let us say the default value as false.

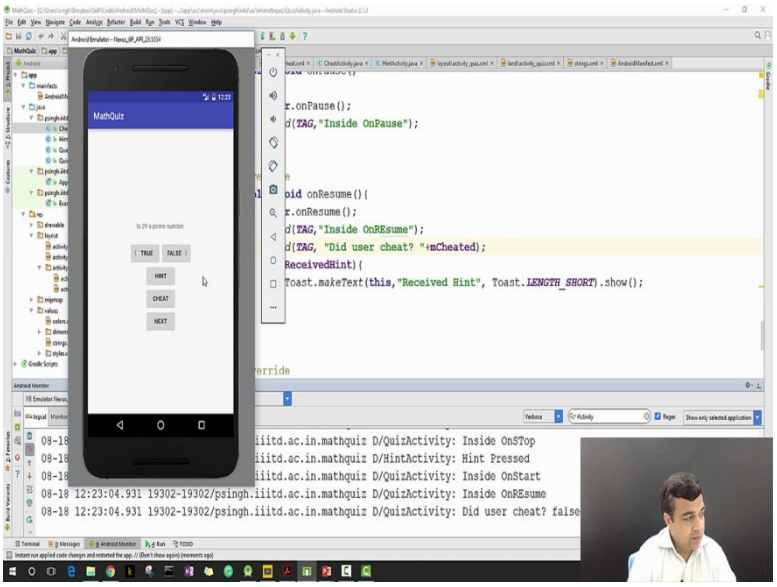
Now, if you try to understand the flow, when our activity gets started it receives an intent takes the data out and gets started. But it wants to return the data, it again uses an intent, puts the data into it and returns it. We want to access this data, we try to get the same value out which we put, so here we are putting a Boolean and we are taking a boolean out.

(Refer Slide Time: 44:25)



Now, let us go to quickly go back to our previous one, Cheatactivity.wascheatshown and data. Hopefully now our program will work fine and one thing that we can do is to may be add another log button which will show us the value of mcheated. So let us add a message to the onResume log.d TAG Did user cheat? Now I will say mcheated, now let us start our application and hopefully we should be able to see data going from parent activity to child activity and then from child activity to parent.

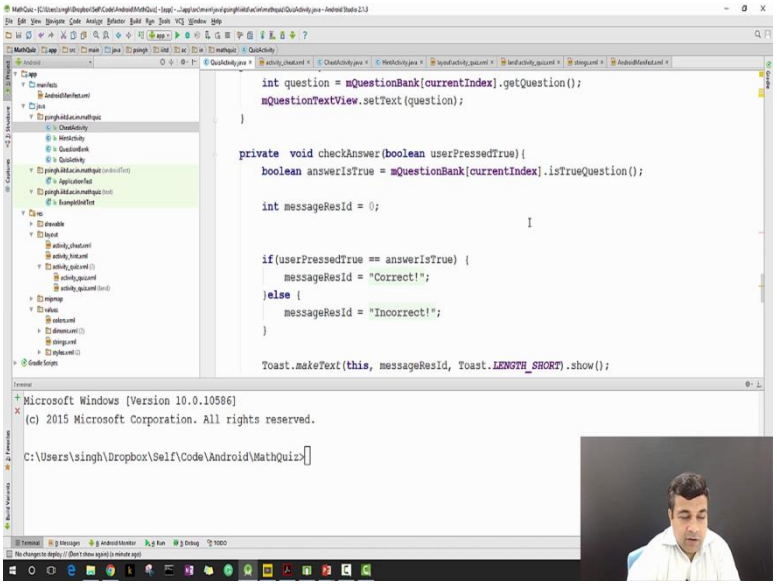
(Refer Slide Time: 46:02)



As you can see initially Did use a cheat is resulting into a false. We can do, everything is fine. Let us press the hint button, come back still we will use a cheat as false, because you got the hint but it is not cheat. Now, let us see let us press the cheat button, go to the cheat we can see

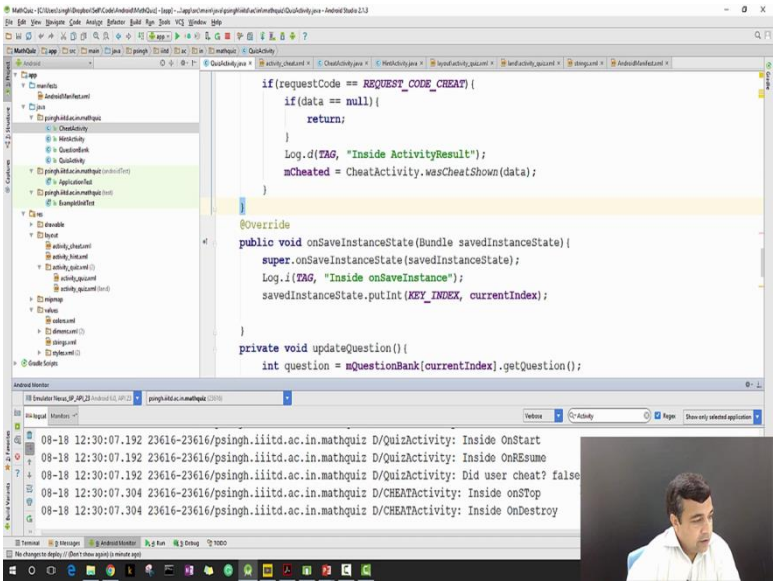
the value received is 1 which was equivalent to the current index that we had seen earlier as well. Now, we will try to go back and see whether we get data back from the Cheat activity to the Boolean and as you can see that it is still showing false, let us try to solve this error.

(Refer Slide Time: 47:23)



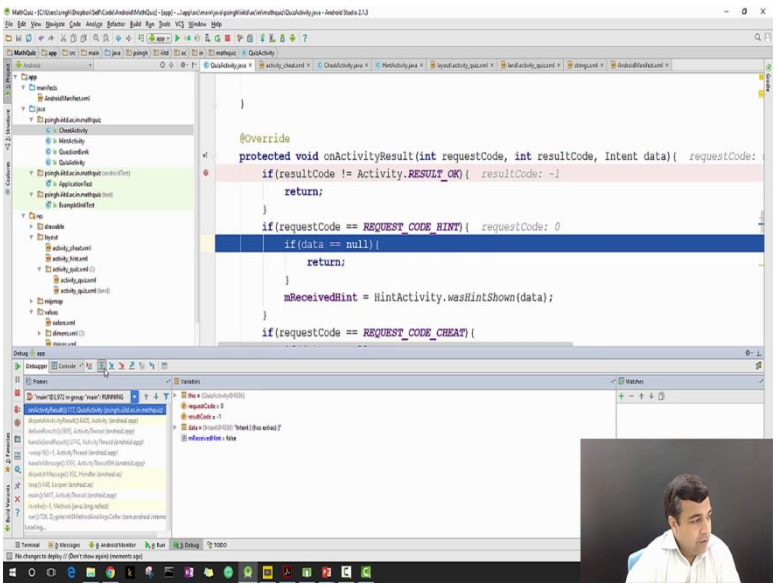
I will put a log message here log.d TAG Inside activityresult ok, so let us try to fix the error and this is also a good test on how do we find out about the errors something which we learned in our last class. So, as you can see that currently even though we are cheating, it is turning a false value. Now, let me try to see that why it is doing it. So, last time we learned about debugging let us use some of those uses. First thing that I am going to do is to put a break point and I am putting a break point at this one because I want to know that whether this method is being called or not and if this method is being called then why our method is not been called because we did not also get this log message. So, let us run our program but this time we will run it in the debug mode.

(Refer Slide Time: 49:05)



To press the debug mode, we start, we press the cheat button, come back. When we come back as you see the execution has stopped, so onActivity result is called that is right. However, onActivity method is saying that the result code is 0 While we are expecting - 1, and because we are expecting a - 1 it will directly go to return, not going to the any of that and that is it and that is why our messages are not getting printed. Ok so, we have found out the error. Let us see what happens when it has hint which was working, so when press hint, press hint come back again our code is stopped.

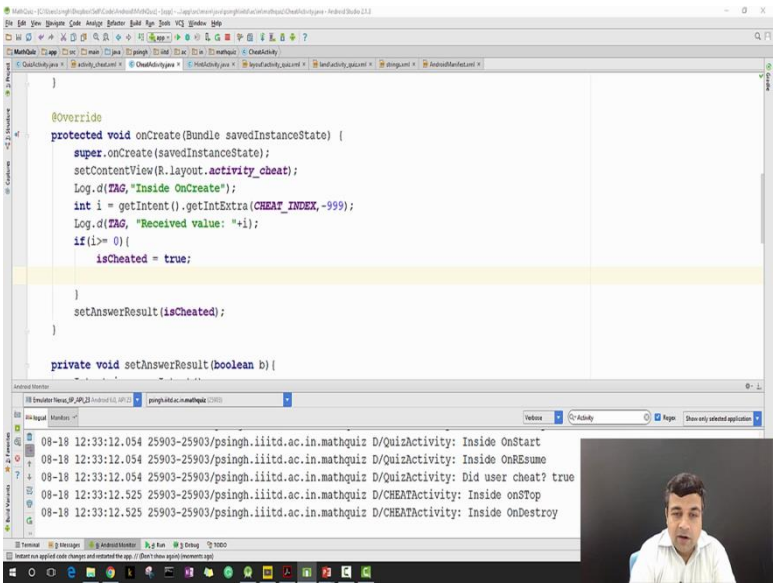
(Refer Slide Time: 50:32)



Ok, we have to press also the showing button. Let us start again, this time we will complete the hint, we will go to the hint, we see the hint we come back and as you see that our result

code is what is being expected. If I stop over it goes into the request code hint because our request code is 0. It will go further and it will set the mReceivedHint. So, as you can see that, when we are pressing cheat our result code is different comes out of 0 value while it should be - 1. So, essentially our result code is not being set correctly in our cheat activity class. So, we have identified it using a simple, single break point. Now let us go to cheat activity and try to see why it is not happening? So, in check cheat activity we came to onCreate we received this value and we are stating our result into this function, but we never called this function.

(Refer Slide Time: 52:26)



If we never call this function then it will not work as you already know. So, let me call setAnswertrue with scheated and let me put it after that. Now I hope that I have corrected my error and I will directly run my program if things do not work then I will go back to debugging again. Like we learned in the last class, debugging is an integral step for coding. So, yes we are back, we can see the log messages. I will clear out all the messages, before we start if that is next so, I tell you to receive a value 1 press cheat with receive the value 1 we come back and yes this time we got the right message.

So as you saw number 1, how we created another activity in our application. Number 2, how we passed the data from one act from the parent activity to the child activity. Number 3, how did we pass the data from the child activity to the parent activity? And in between I also showed you briefly how to debug your program when you are getting an error. Now in the next video we will actually write more code and fix our math quiz program so that instead of just displaying a simple message, we actually calculate and come to know about the right

answer when we cheat and we will also change our toast in the parent activity to reflect that the user has cheated.

Thank you.