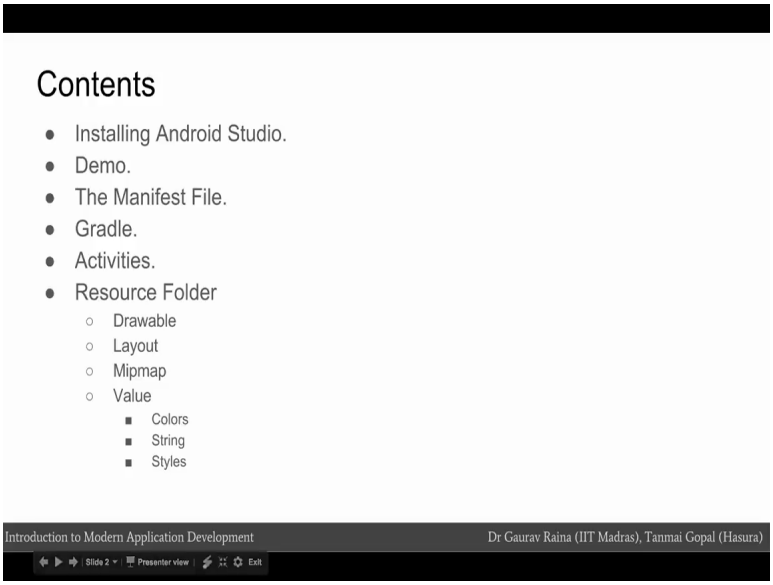
Introduction to Morden Application Development

Dr. Gaurav Raina
Prof. Tanmai Gopal
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

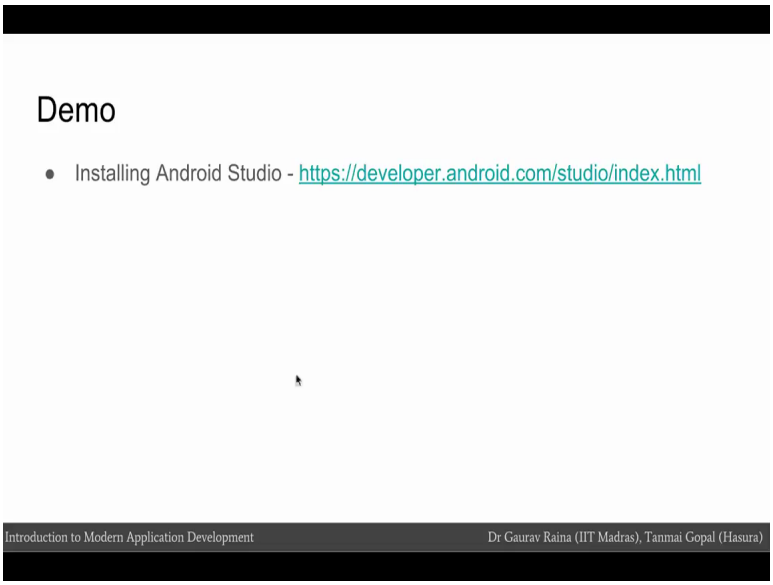
Module – P12
Lecture – 30
Getting started with Android Application Development

(Refer Slide Time: 00:07)

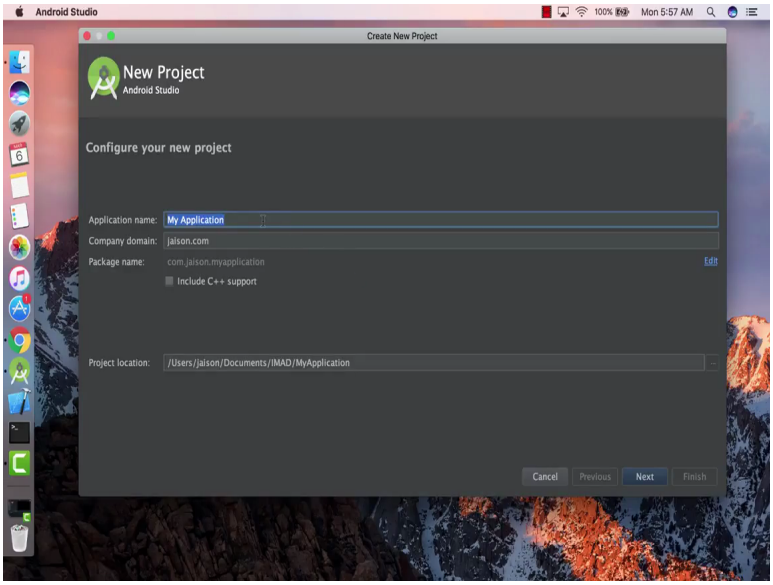


Hey everyone, in this module we are finally, going get started with android application development. We would begin by installing android studio and then having a quick demo of what a simple app looks like and then we will finally move on to exploring and getting a better understanding of the different components that make an android app.

(Refer Slide Time: 00:23)



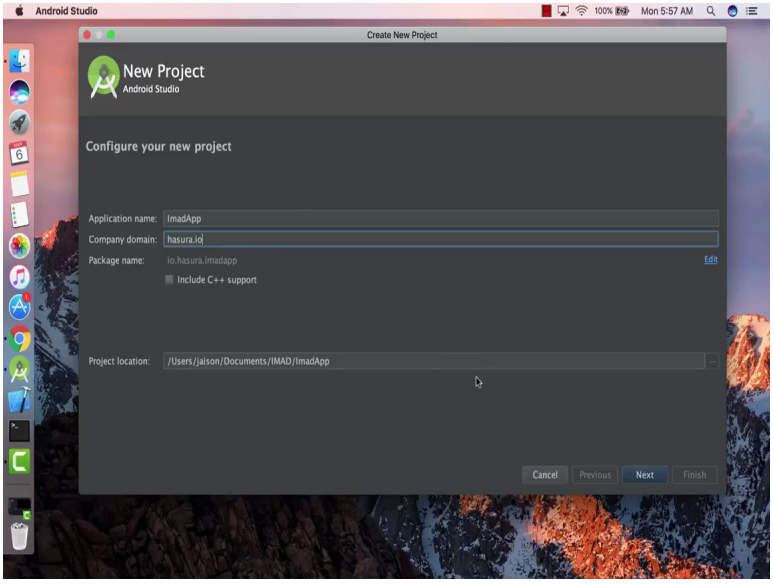
(Refer Slide Time: 00:27)



Let start by clicking on the link provided in this slide and downloading android studio on to our computer. Basically if I click on download android studio and follow the instructions that come on the screen.

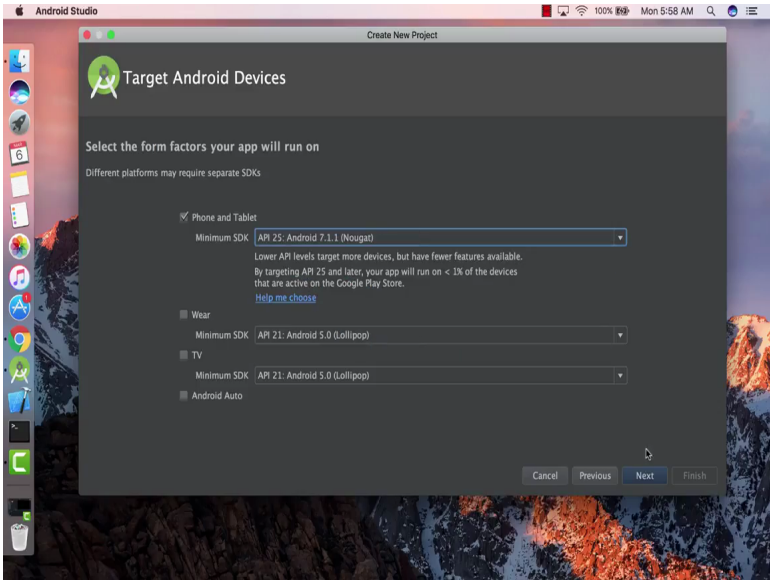
I already have android studio installed in my laptop and hence I am not be downloading it after you have done installing android should be on your laptop open it up and create a new project. So, in this screen you first ask to choose a name for your application you can choose this to be anything.

(Refer Slide Time: 01:12)



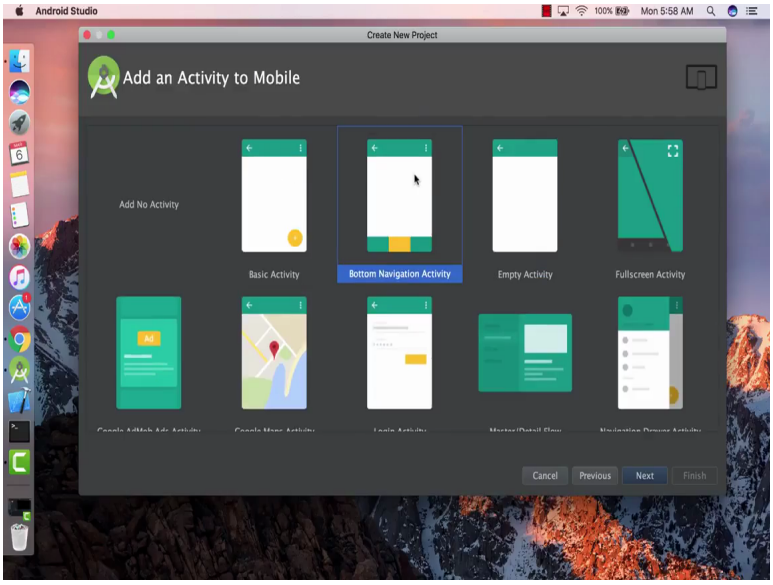
In my case I am going to choose it Imad app, the company domain is basically something that android studio uses to create a package name for your app which is basically a unique identifier that is used to identify your app in the Google play store. It can be anything and usually the package name follows the reverse naming convention. So, in this case if you if you company domain is JSON dot com or let us say Hasura the pack name would be i o dot Hasura dot Imad app choose a project location click on next.

(Refer Slide Time: 01:52)



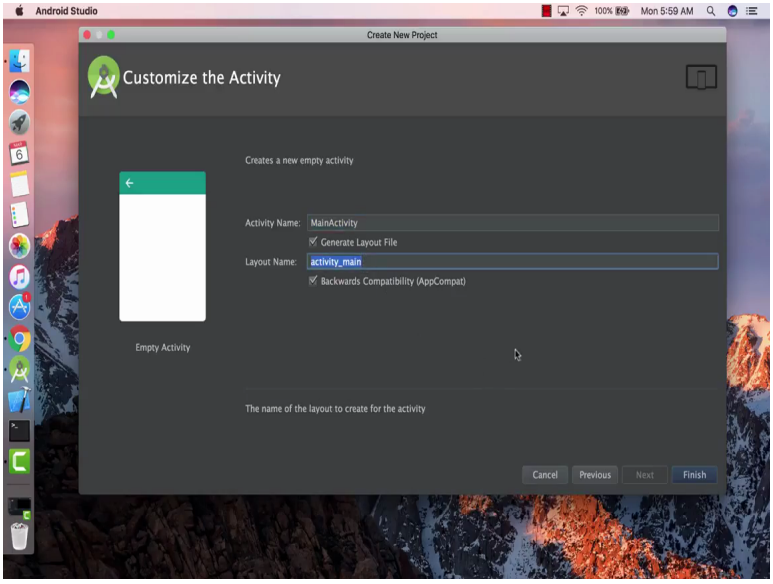
This screen is basically asking you to choose what type of devices android devices that you support and the minimum SDK for each of those devices. We are going to be developing an app for the phone and tablet the minimum SDK can also be chosen from this dropdown list which starts all the way from android 2.3 until android 7.1.1. Another handy feature in this is that then you select a certain android version android studio also tells you the approximate percentage on devices that your app would be running in or approximate percentage or devices that your app would be supporting which is on the app the Google play store.

(Refer Slide Time: 02:44)



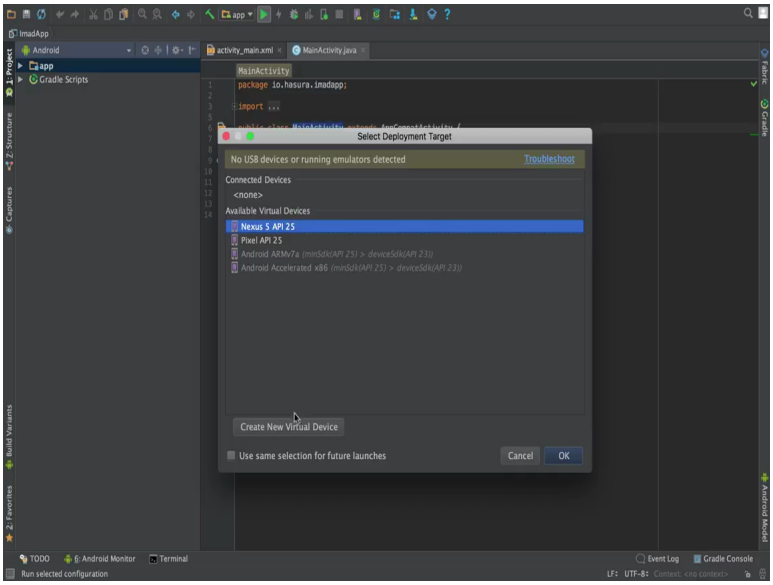
In our case let us choose the latest android version click on next this screen is for you to select a starting activity for your app it is a good practice or it will be it would be interesting for you to actually go and test out each of these different type of starting activities to see how android code will set up in each of these for the case of this module let select empty an empty activity.

(Refer Slide Time: 03:05)



That the name of the activity be main activity ensure that generate layout file and backwards compatibility app compact check boxes are checked and the layout name is activity underscore main. You can change this to anything you want, but for now you can leave it as it is, click on finish and let android to build or create the project for you.

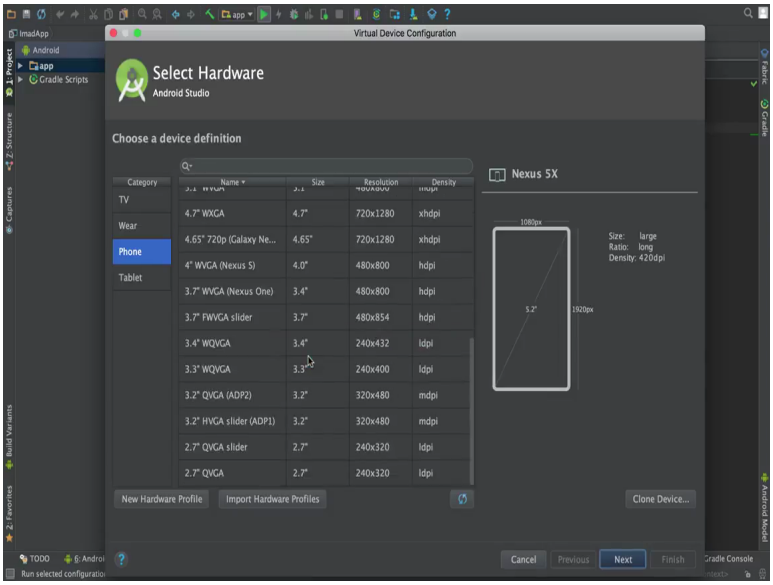
(Refer Slide Time: 03:27)



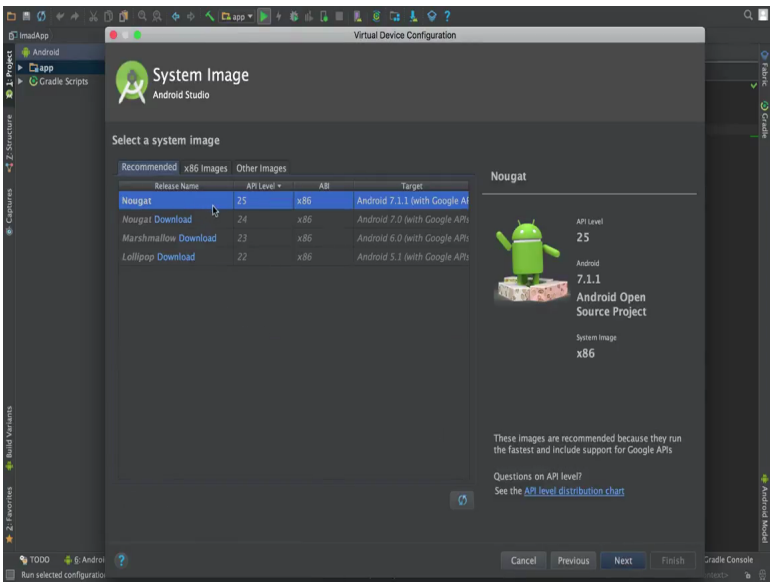
Once android studio is done setting up our app you be presented with the following screen ensure that android is selected from this list if you do not have this panel just click on the project icon on the left and you should be you should get this. Now let us just run

the app and so by default when you create a project android studio sets up a HelloWorld app for us. So, let just run that app and see how it looks. When you click on the run icon you presented you would be presented with this screen which is ideally asking you to select the device on which you would like to run this app. If you have an android device you will have to enable developer settings on it you have to Google to figure out how to enable developer settings on your specific phone if you do not have an android device you can always create a virtual device using android studio.

(Refer Slide Time: 04:24)

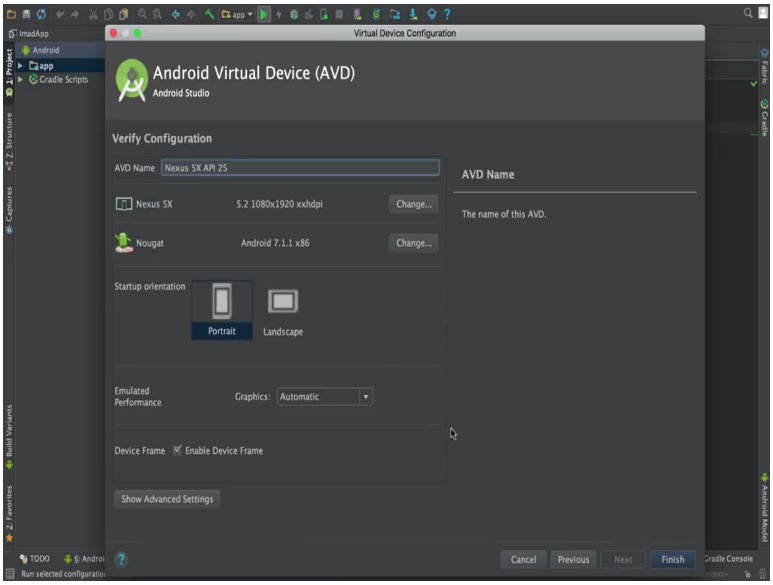


(Refer Slide Time: 04:29)



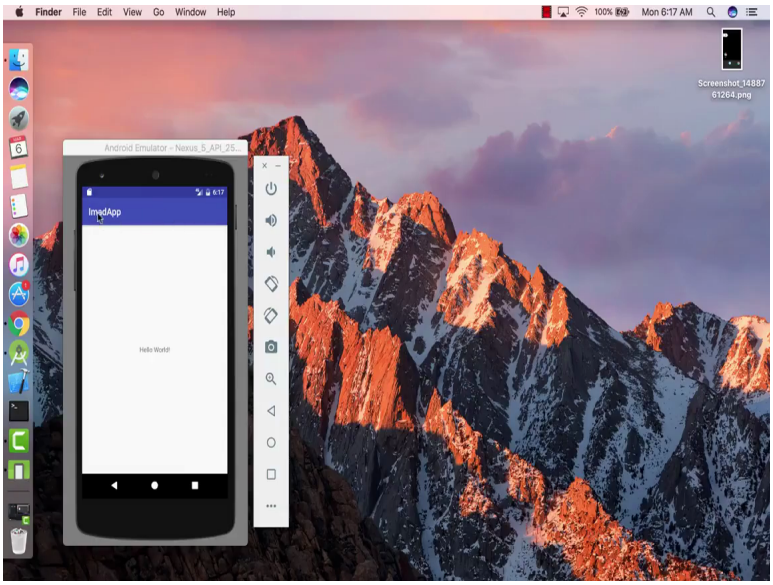
For that just click on the; create new virtual device bottom select the device that you would want click on next.

(Refer Slide Time: 04:32)



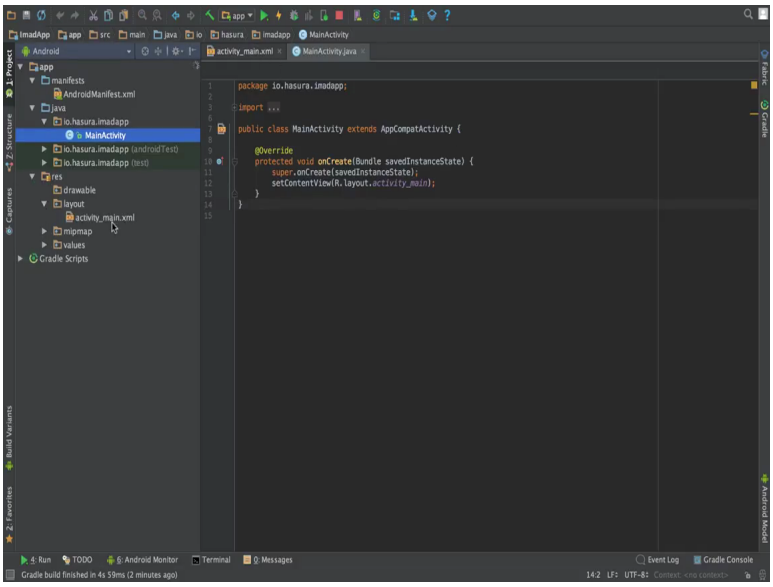
Select the android version that you want to run on and click next and give it a name and then finally, click finish to create the virtual voice. I have already created an access five which runs on API 25 and I will be selecting that to run my it gives android studio something to setup the virtual device and run the app on.

(Refer Slide Time: 04:55)



As you can see the emulator is quite handy tool gives you features to rotate the screen to capture snapshots to mimic the volume up and volume down button and also such things. So, now, our app is run it is a very simple app it is a hello world app which is displays a text saying hello world in middle of the screen. It has an action bar or a tool bar with the name of our app on it yes now I have look at how this app is working and what all are these files are and what they do.

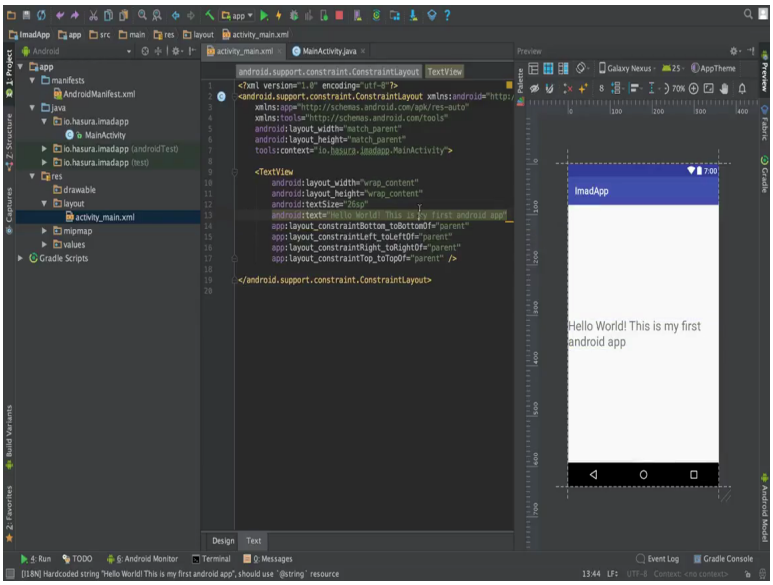
(Refer Slide Time: 05:37)



First navigate the main activity to a java file which should be open by default in your android studio in case it is not navigate to app java package name and main activity. As you can see it has one method called on create where in we setting the UI for this activity with a method called set content view and by sending a layout file called activity underscore main activity underscore main is an XML file and it is one of the files that is provided by the resource manager which is one of the functionalities of the java API framework.

The activity underscore main file should also be opened by default in your android studio in case is not you can find it by navigating to the resources directory clicking on layout and activity underscore main.

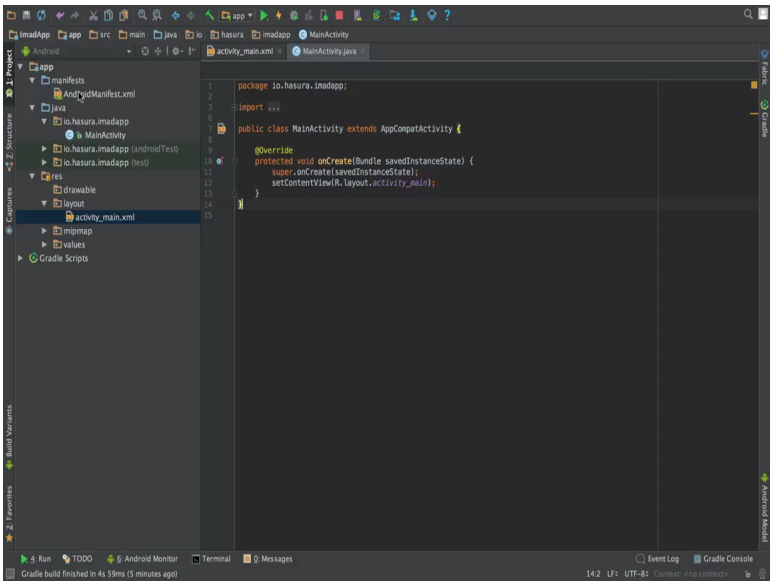
(Refer Slide Time: 06:32)



As you can see Android studio has a pretty good interface for designing UI to your apps. In this case we have hello world text view put in at the centre of the screen. Let us see how this looks in XML. To get this preview you have to click on the right the activity underscore main file has constraint layout set as its root view where in the layouts within height as set to match parent here match parent mean that it would take width and height of this of the device that it is running on inside this constraint layout we have a text view which is the widget that shows text.

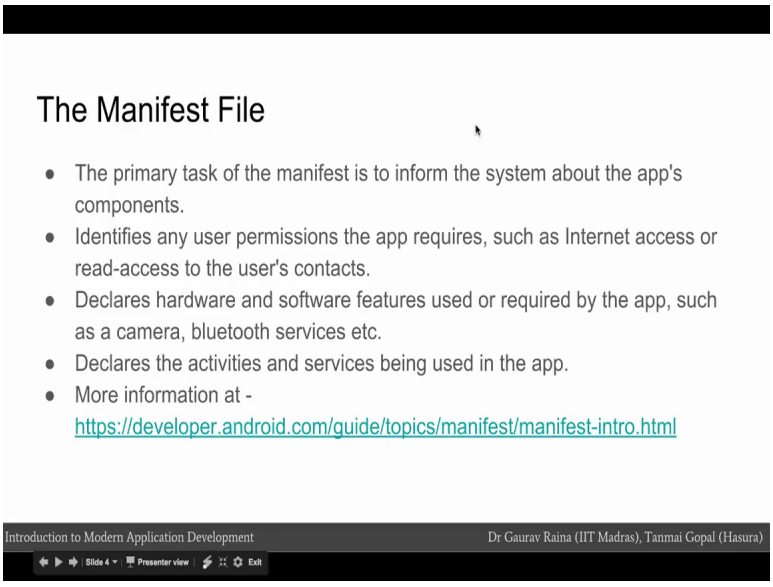
The important thing to notice here is the width in height of the text view which is set to wrap content this means that the width and height of the text view would wrap its content and resize it cell based on the content that it is showing. For example, right now this is how the text view looks like in case I would change the text to hello world this is my first android app as you can see the width of the text u has changed. Similarly if I were to change the text size to 20 or 26 once again yeah 26 as you can see the height of the text we has changed, this is because it is researching itself to wrap its content.

(Refer Slide Time: 08:20)



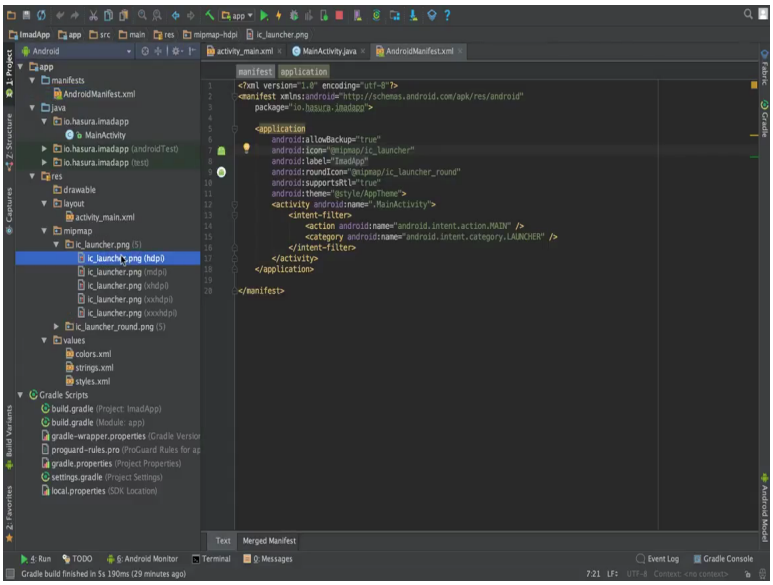
Now, before we go on to manipulating the UI achieving the UI lets understand what each of these files do. First we have the android manifest the android manifest, as a name suggest informs the system about the apps components.

(Refer Slide Time: 08:32)



It identifies any user permissions the app requires such as internet access or read access to the user's contacts it also declares hardware and software features you use the required by the app such as the camera or Bluetooth services it also declares.

(Refer Slide Time: 08:59)

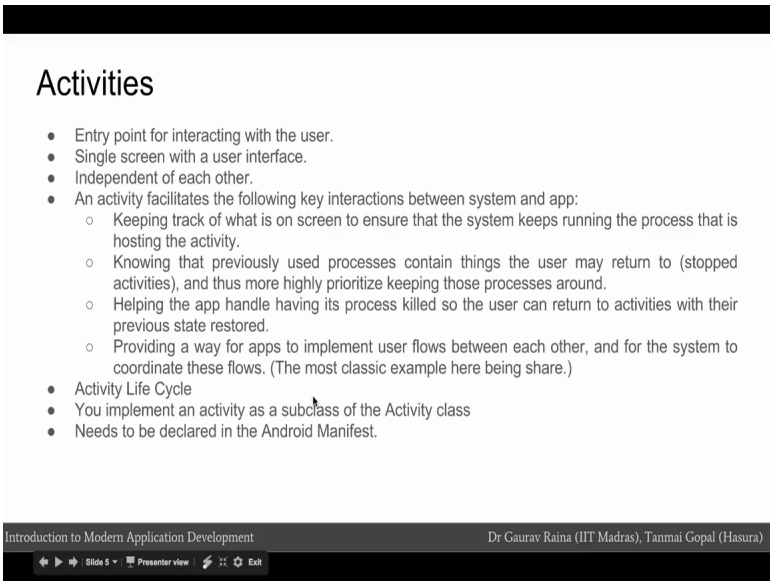


The activities and services being used in the app for more information about the manifest file do is the link provided in the slide there are a few things to be noted in our android manifest.

Firstly the package name of a is specified in the manifest we have also declared main activity as an activity in the manifest. Now since every app needs starting point this two needs to be declared in an manifest we have assigned or only activity the main activity as a starting point this is done by setting the two intent filters one with the action main and the other one with the action launcher with the category launcher. Some of the other things in the manifest are allows backup which we have said to true it also defaults to true this essentially means that the android OS will periodically backup (Refer Time: 09:40). The icon is the icon used by our app on the home screen be setting a PNG file to be the icon which is inside the Mipmap directly under the resources as you can see android studio is also provided the same image in different sizes.

By setting supports RTL to true this best friend that are app supports right to left layouts this is used for languages like Arabic we will also setting a default theme for the whole app as app theme will get to this in a bit.

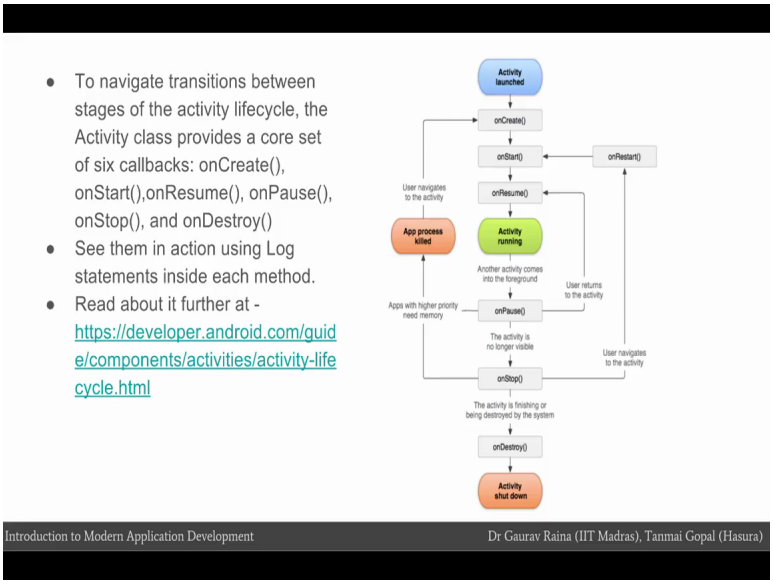
(Refer Slide Time: 10:17)



Let us now talk about activities an activity is an entry point for interacting with the user it is a single screen with the user interface and app can have multiple activities and each activity is independent of each other. And activity facilitates key interactions between the system and the app like it keep track of which processes to run based on what is being shown to the user it also tracks the previous stop processes which the user may return to and gives them higher priority.

Activities help the app handle user state and also provide a way for apps to implement of flow between each other. One of the most important things to know about activities is the activity lifecycle.

(Refer Slide Time: 11:03)



As in user navigates in an out of your app the activity instances undergo transition in their life cycle states the activity class provides call backs to allow the activity to know that is status changed to navigate transactions between stages of the activity lifecycle the activity class provides a set of 6 call backs on create on start on resume on pause on stop and on destroy.

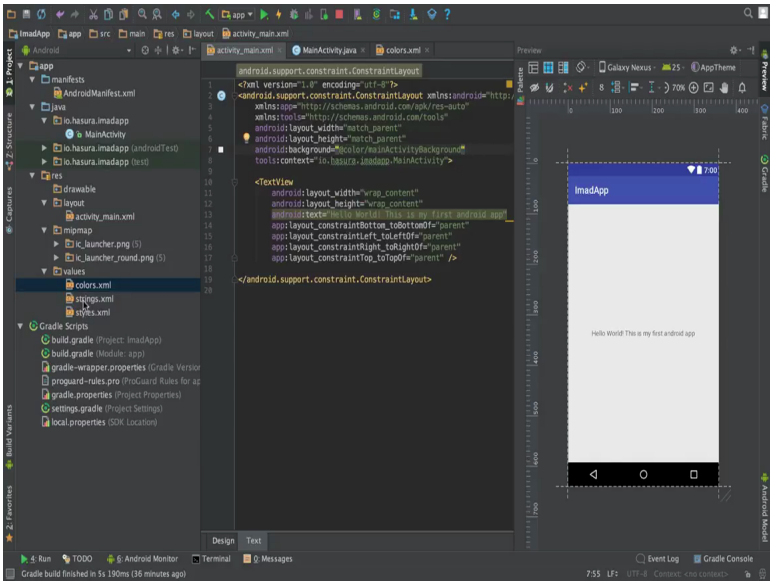
The image will provide a better understanding of these call backs. The on create method is called right after the activity is launched and created you usually put startup logic for the activity in here things that should be implemented only once exciting the view and binding them like we are done in our main activity class. This method receives the parameters saved instance state which is a bundle object containing the activities previously saved state if the activity has never existed before the value of the bundle object is none. And then comes the on start call back which the system invokes when the activity enters the started state this call makes the activities visible to the user and the activity becomes interactive this is where the app prepares for the activity to enter the program.

When the activity enters the resume state it comes to the foreground and then the system invokes the on resume call back, this is a state in which the app interacts with the user the upstairs in the state until something happens to take focus away from the app. This can be receiving a phone call of the user navigating to another activity of the on the

device screen turning off the on pause method is called as the first indication that the uses leaving your activity. The on pause method is called as a first indication that the user is leaving your activity, on pause is usually use to pause operations such as animations and music playback that should not continue while the activity is in the pause state.

When your activity is no longer visible to the user for example, when a new activity covers up the entire screen it enters the stopped state and the system calls the on stop call back the system may also call on stop when the activity has finished running and is about to be terminated the on destroy, the on destroy is called before the activity is destroyed this is the final call that the activity destroys.

(Refer Slide Time: 13:37)



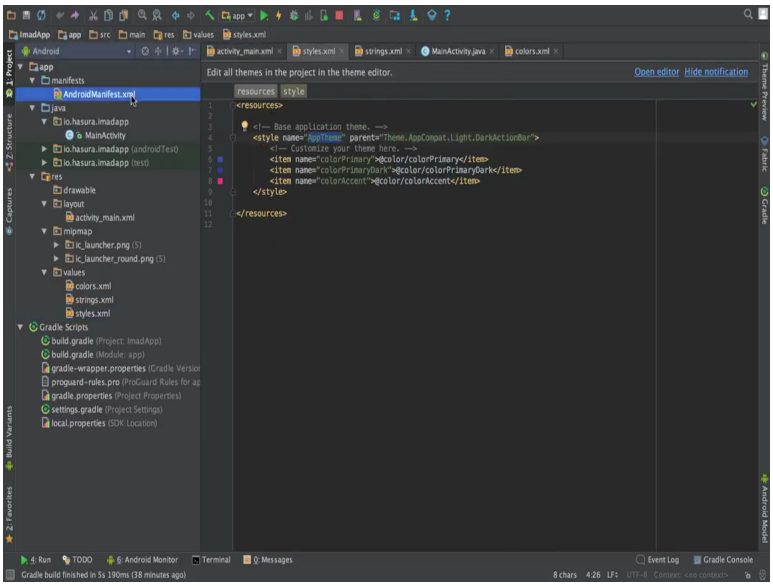
The resource directory holds all the files are provided by the resource manager the drawable directly is where you put all the images to be used in the app the Mitmap is where you include the app icons the layout file has all. The layout XML files we used in activities another views then values directory is what we used to externalize value such as color X codes strings and styles in a project.

So, example the colors dot XML file is where we give names to all the colors X code. So, that it becomes easier to access and changed later let us add another color to this and call it main activity background we set the color to be differentiate of white. So, now, we can go to activity underscore main XML file go to a constraint layout and set the

background to be at the color that which is added and as you can see the color has changed next is a string file. The string file is also very important file and it is highly recommendedly use this string files over loosely writing done strings for the views in our app.

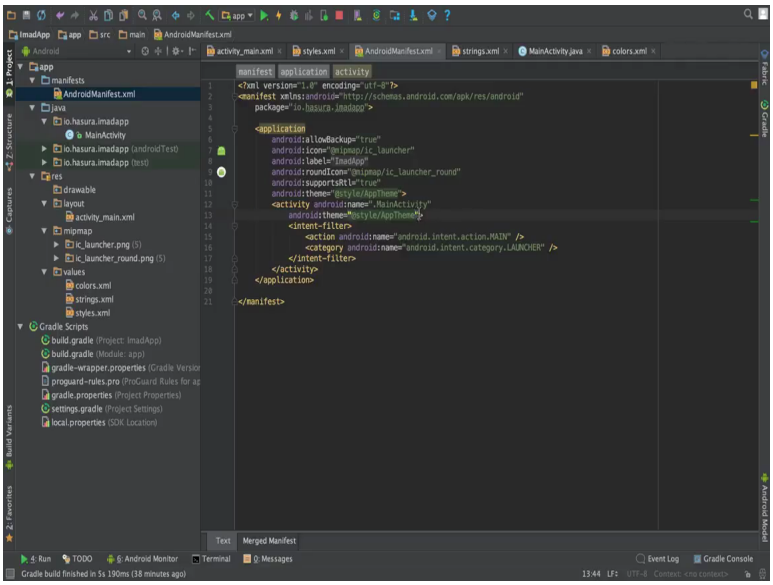
So, example what we have done here is should not be done instead we should copy this string and place it in the strings files.

(Refer Slide Time: 15:08)



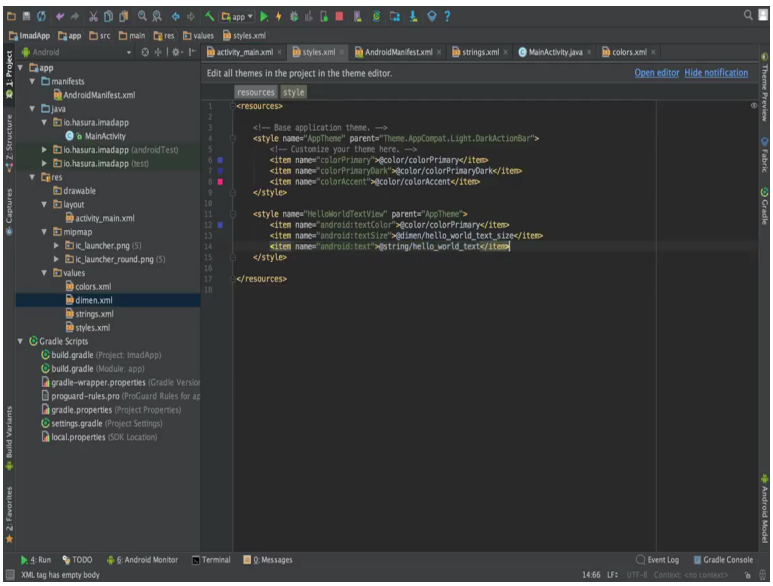
Let us do that just call it hello underscore world underscore text and. So, now, if we go to the they go fuses styles are another feature in android there in to avoid code repetition you can create a specified styles for app or for the different widgets in your app. In our case we have created a style called app theme whose parent is theme dot app Compat dot light dot action bar dot action bar. It is because of this theme that we get the action bar in our app by default with the dark blue color I will just show that to you, this action bar that we get in our app by default when we run the app is because of the theme that we are specified for our app in our android manifest.

(Refer Slide Time: 16:04)



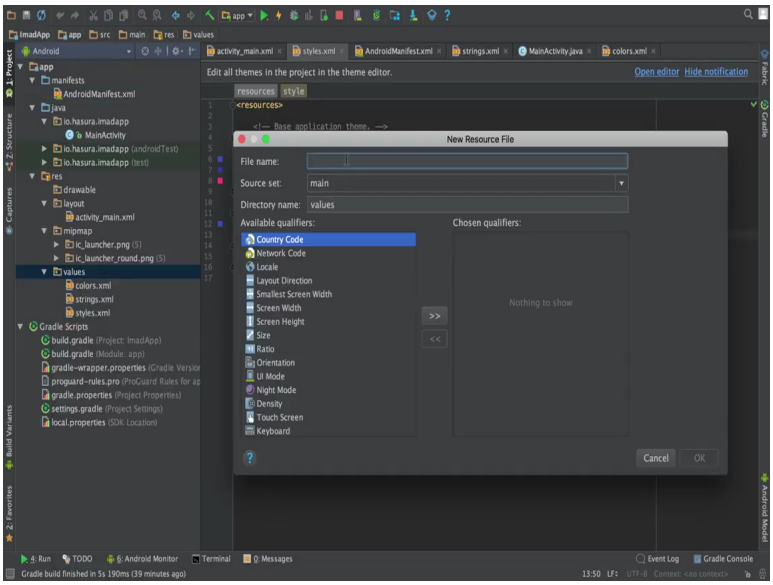
By setting it here we set app theme as a default theme throughout our application, we can also specify themes specifically for activities.

(Refer Slide Time: 16:26)

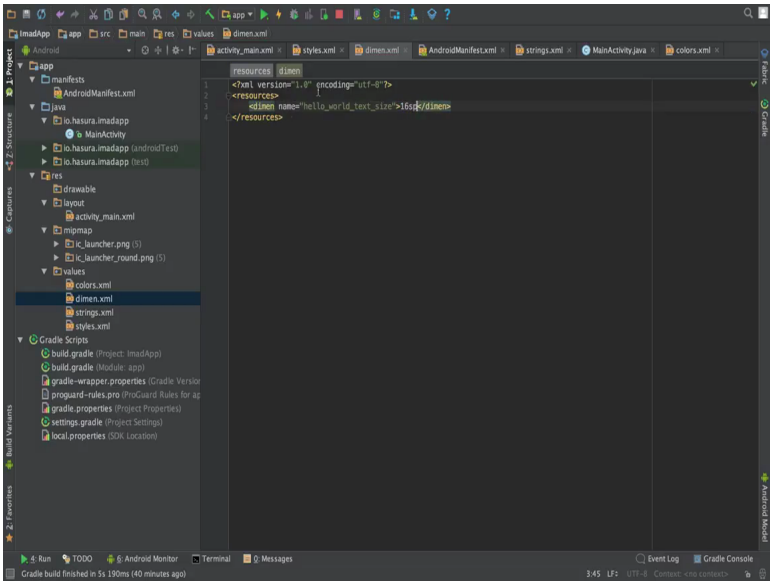


Let us create another style for a text view as call it HelloWorld, hello world text view and its parent be app seen that is the main text color to be at color primary name size 16 SP here if a size as well we can do another thing.

(Refer Slide Time: 17:33)



(Refer Slide Time: 17:43)

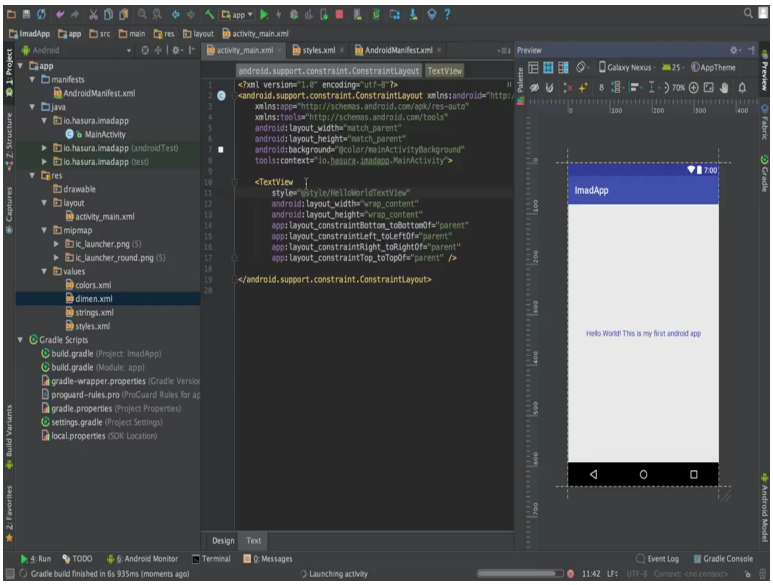


We can have another value resource file call it dimen; dimensions and do the same thing here call it dimen name and read as 16SP and change this here to you (Refer Time: 18:03).

So, we made a new dimensions file and added a new dimension to it call hello world text size and use that. Doing this is very important necessary it might seem unnecessary at this upon a time, but then you building huge applications, there will be code repetition everywhere and doing this prevents that and also give you a centralized placed place

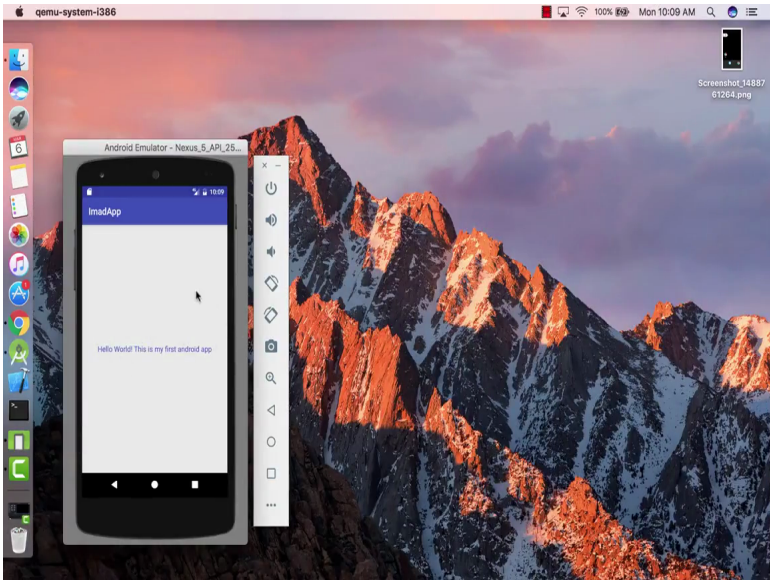
where all the values and where all the heart coded values are placed. This helps in changing things later. We can also add things like layout height and width, etcetera, but I am going to live that for the time being. So, let us go to our text view as also I had a string here, yeah.

(Refer Slide Time: 19:03)



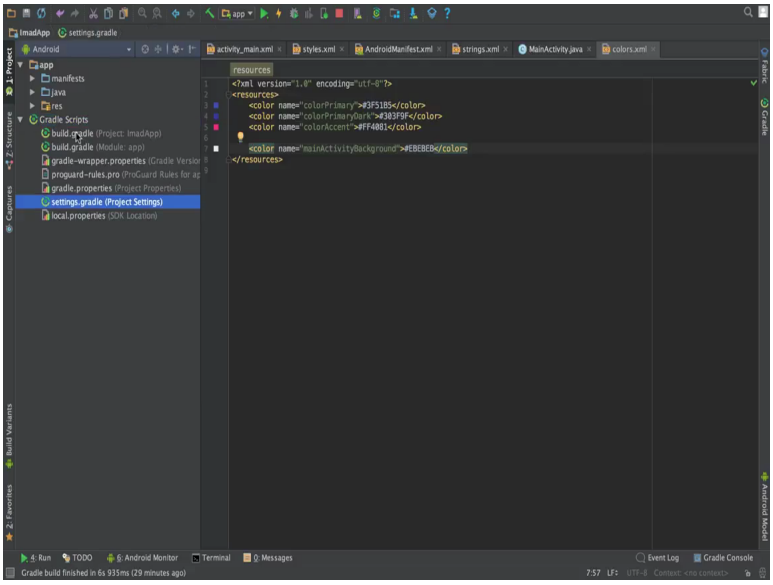
Now, let us go back to our android underscore mean XML file let us take off this and then let say style let say style. So, now, as you can see this change is reflected, let us run our app and see.

(Refer Slide Time: 20:05)



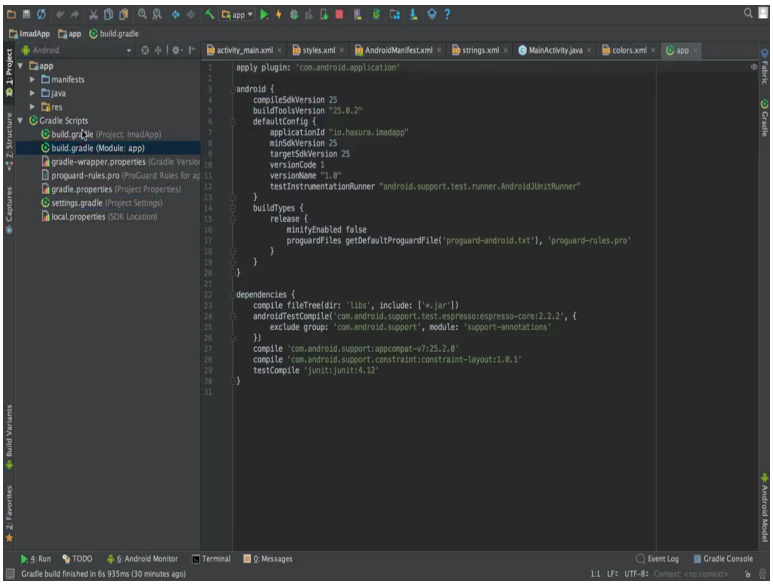
Yeah, as you can see the color of the text, text is changed finally we have Gradle which is a custom build to use to build APK files. It manages dependencies and provides custom built logic, Gradle introduces a Grue based domain specific language DSL.

(Refer Slide Time: 20:11)



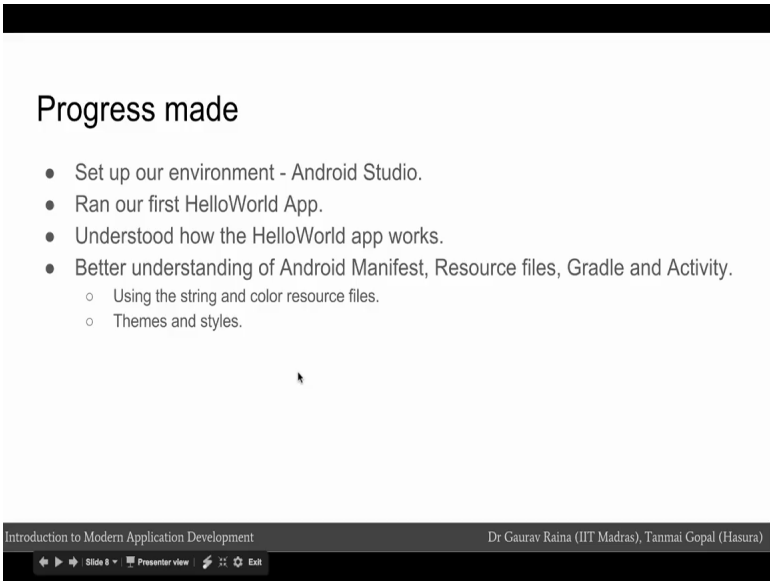
In short typically we have 3 Gradle files present in an android project we should be two build of Gradle files one at the app of module level and in other one at the project level and finally, a settings dot Gradle files.

(Refer Slide Time: 20:34)



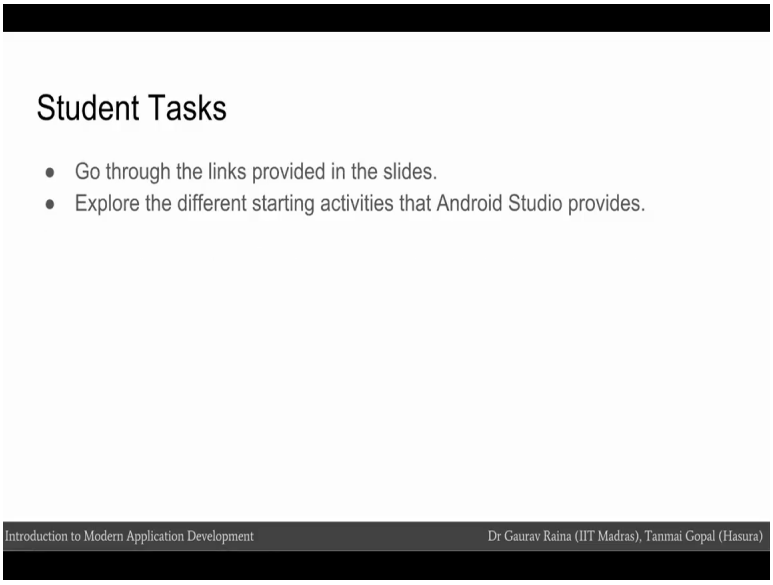
We will mostly be using the app level built of Gradle file we will get to this in more detail later. With this we come to the end of our module.

(Refer Slide Time: 20:46)



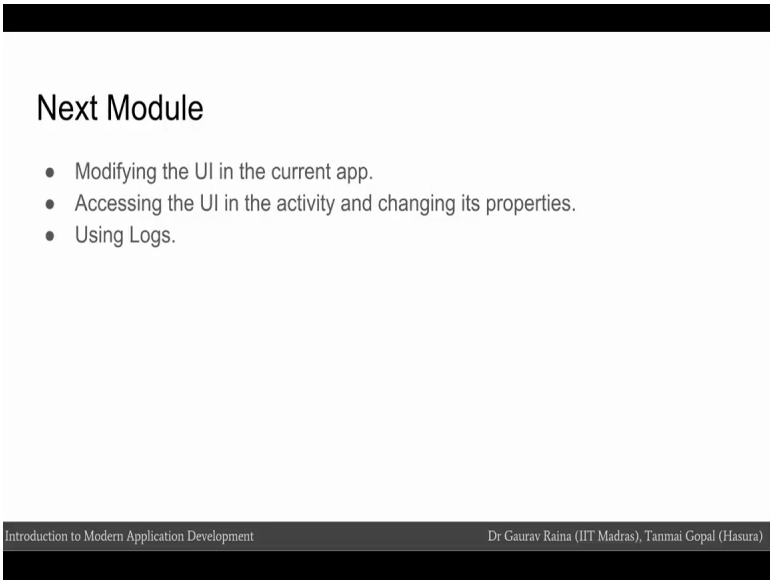
In this module we have setup our environment to get started with android application development we have downloaded and installed android studio on our systems. We ran our first hello world mobile android app we understood how this hello world app works is got an better understanding of android manifest resource files, Gradle and the activity. We have also used this string and color resource files in an android project and explode using themes and styles.

(Refer Slide Time: 21:21)



If you serious about android application development then it is important that you go though the links provided in these slides and get a better understanding about the concepts that we are explode here today. If you remember when we created a project we have presented with the option of choosing a type of a starting activity of android studio we are chosen to go with an empty activity. It will be good exercise for you to explore the other activities provided by android studio understand how they work and how the UIs created and why they difference from each other.

(Refer Slide Time: 21:57)



In the next module, we will modifying the UI of the current app it also be looking at how to access this UI elements in the activity and to modify the properties base on user interruption and finally, we will also be looking at logs.