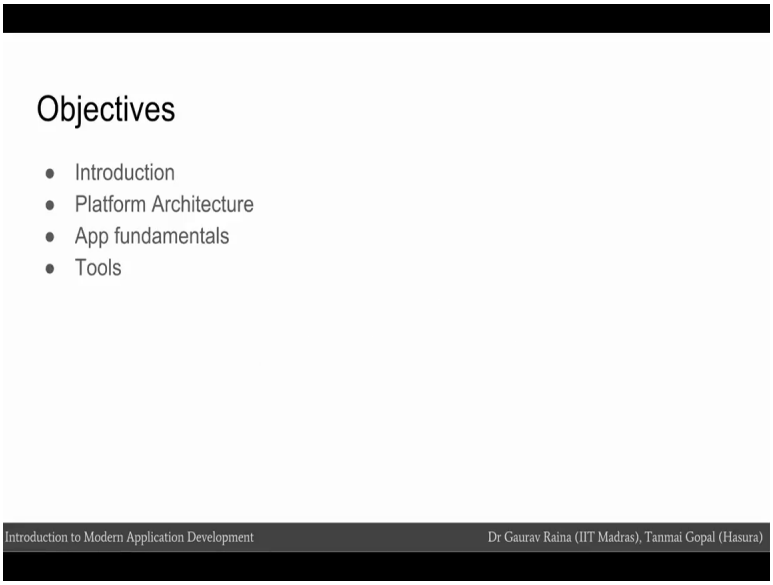
Introduction to Morden Application Development

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

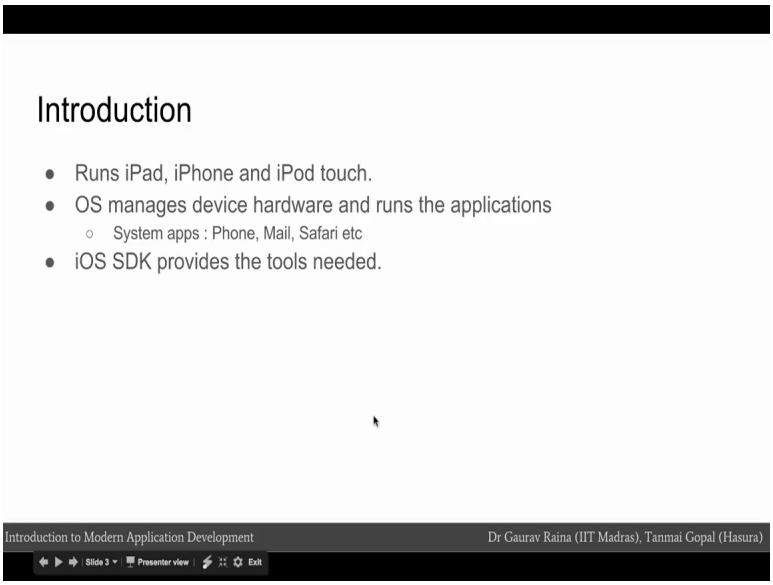
Module – 19
Lecture – 34
Introduction To iOS

(Refer Slide Time: 00:01)



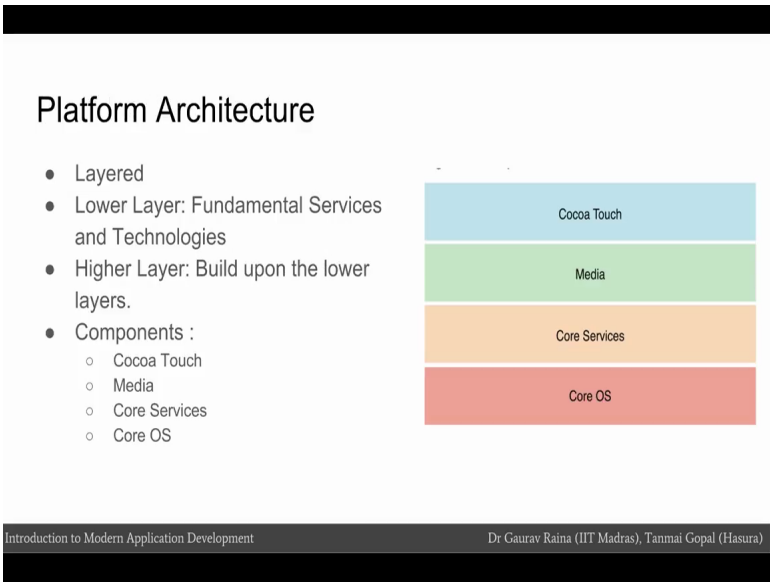
In this module, we are going to explore the iOS ecosystem and its platform architecture. We are then going to look into a few of the iOS app fundamentals and finally, list down the tools required to built an iOS app.

(Refer Slide Time: 00:14)



IOS is the operating system that runs on iPad, iPhone and iPod touch devices the operating system manages the device hardware and provides the technologies required to implement native apps. The operating system also shifts with various system apps such as the phone, mail, safari etcetera. The iOS software development kit or the iOS SDK contains the tools and the interfaces needed t o develop install run and test native apps.

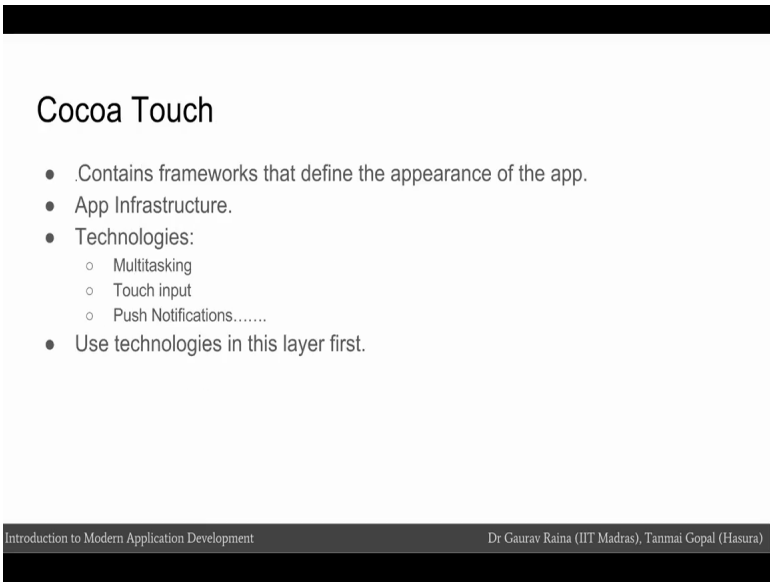
(Refer Slide Time: 00:41)



At the highest levels iOS acts as an intermediary between the underline hardware and the apps you create apps communicate with the hardware through a set of well defined

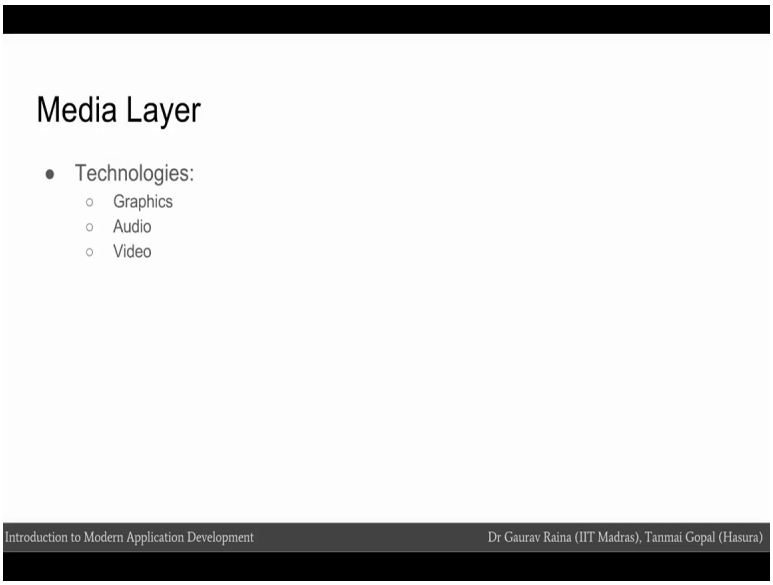
system interfaces and not directly these interfaces make it easy to write apps that work consistently on devices having different hardware capabilities. The implementation of the iOS technologies can be viewed as a set of layers which is shown in the figure the lower layer contains fundamental services and technologies whereas, the higher level layer builds upon the lower layers and provides the more sophisticated services and most sophisticated technologies.

(Refer Slide Time: 01:16)

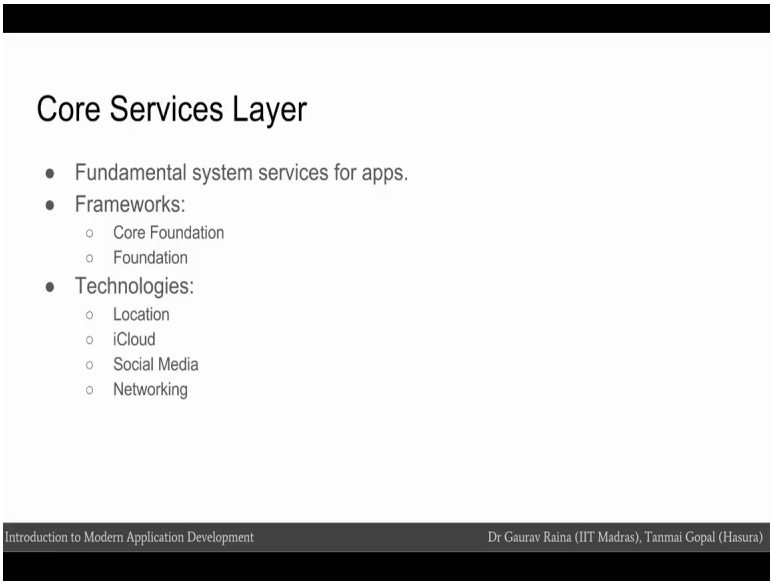


The Cocoa touch layer contains K frameworks for building IO graphs, these framework defines the appearance of your app we also provide the basic app infrastructure and support for key technologies such as multitasking touch based input push notifications and many high level systems services and designing your app you should investigate the these technologies first basically we should investigate the technologies and this layer first to see if the (Refer Time: 04:41) need before going to be other layers. The media layer contains the graphics audio and video technologies you used to implement multimedia experiences in your app.

(Refer Slide Time: 01:43)

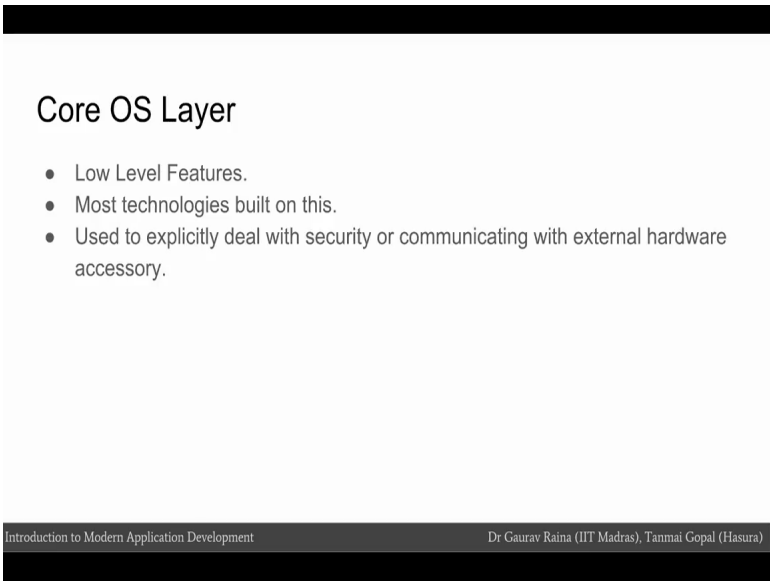


(Refer Slide Time: 01:57)



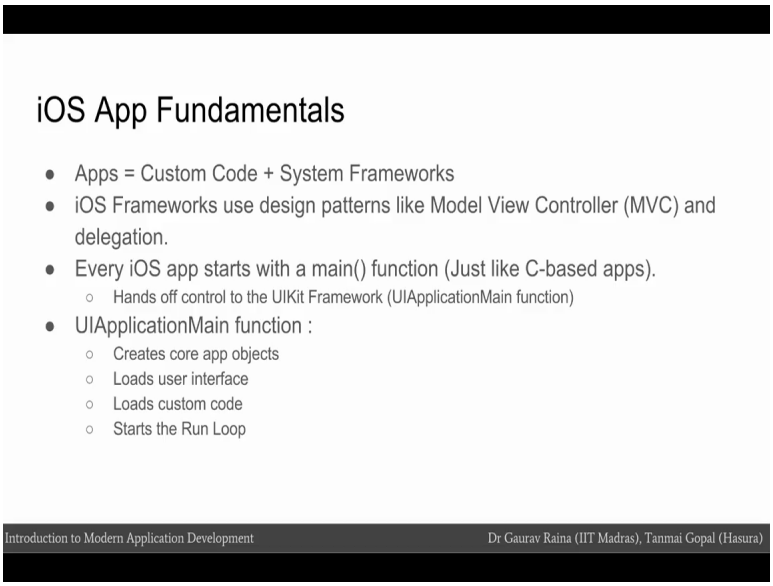
The technologies in this layer makes it easy for you to build apps that look and surrounded. The core services layer contains fundamental system services for apps like the core foundation and foundation framework this layer also contains individual technologies to support features such as location iCloud, social media and networking.

(Refer Slide Time: 02:13)



The Core OS layer contains the low level features that most other technologies are built upon. If you do not use these technologies directly in your apps they are most likely be used as the other framework we use the framework in this layer in situation where you need to explicitly deal with security or communicating with an external hardware accessory.

(Refer Slide Time: 02:32)

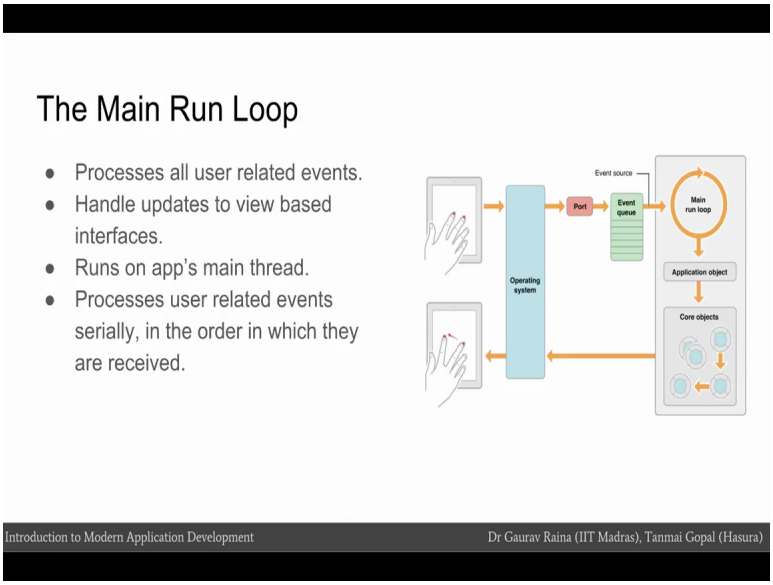


IOS apps are sophisticated interplay between your custom code and the system frameworks, the system frameworks provide the basic infrastructure that all apps need to

run and you provide the code required to customize the infrastructure and give the app that looking field that you want or based on your business or be (Refer Time: 02:53) the color the UI the UX and other such thing iOS apps are c based apps and as every c based apps the entry point of an iOS apps is the main function. What is different is that for iOS apps you do not write the main function yourself instead the id that we use which is x code creates this function as part of your basic projects.

The only thing to mention about the main function is that its job is to hand control off to the UIKit framework where the UI application main function handles this process by creating core objects of your app. It also loads your apps user interfaces calls your custom code and you know so that you can have a chance to do some initial setup and puts the apps run loop in motion, so, the only pieces that you have to provide other views and the custom initialization code.

(Refer Slide Time: 03:40)

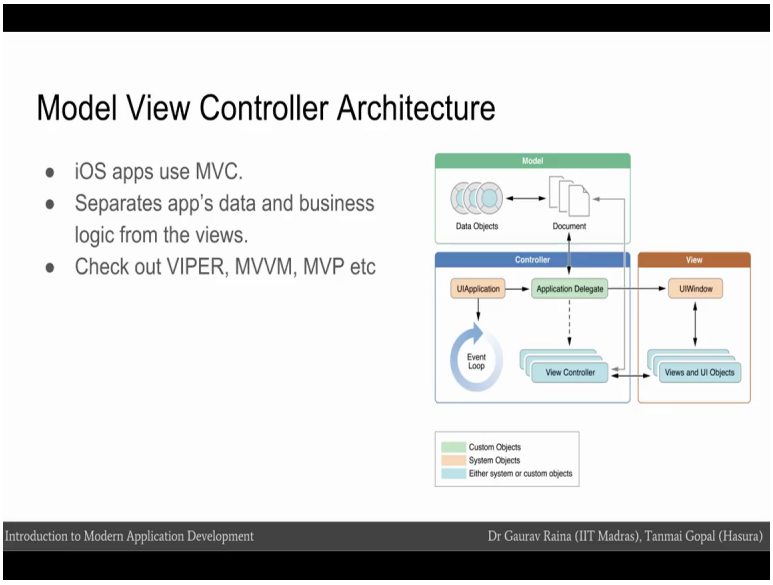


And apps main run loop processes all user related the events. The UI application object success, the main run loop at launch time and uses at 2 process events and handle updates to view based interfaces main run loop executes on the apps main thread.

This behavior ensures that user related events or process serially in the order in which they are received as a user interacts with the device events related to those interactions are generated by the system and delivered to the app by special port setup by the UIKit. Events are cure internally by the app and dispatched one by one to the main run loop for

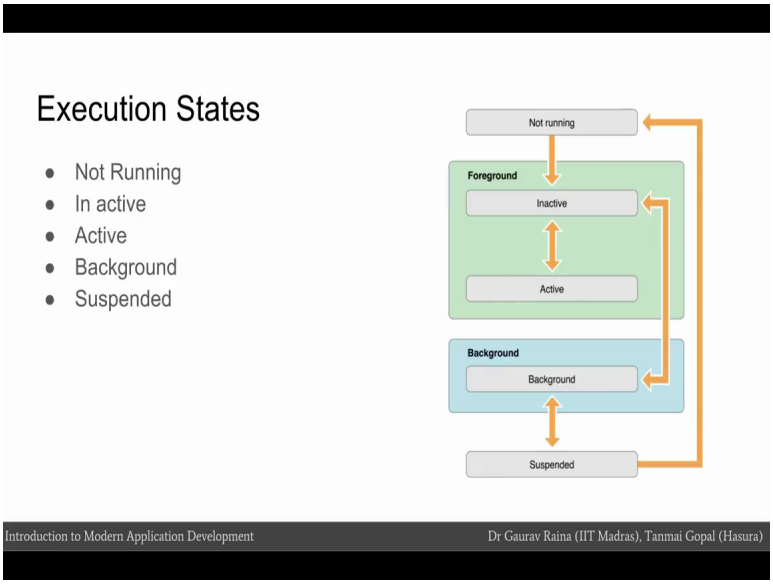
execution. UI application object is the first object to receive the events and make the decision about what needs to be done attach event is usually dispatched to the main window object which in turn dispatches it to the view in which their touch occurred.

(Refer Slide Time: 04:33)



IOS apps use a model view controller architecture what this pattern does is that it separates the app data and business logic from its view. They are also exist other architecture like MVVM which is model view, view model and VP which is model view presenter and viper which also through useful based on the type of app you are building.

(Refer Slide Time: 04:54)



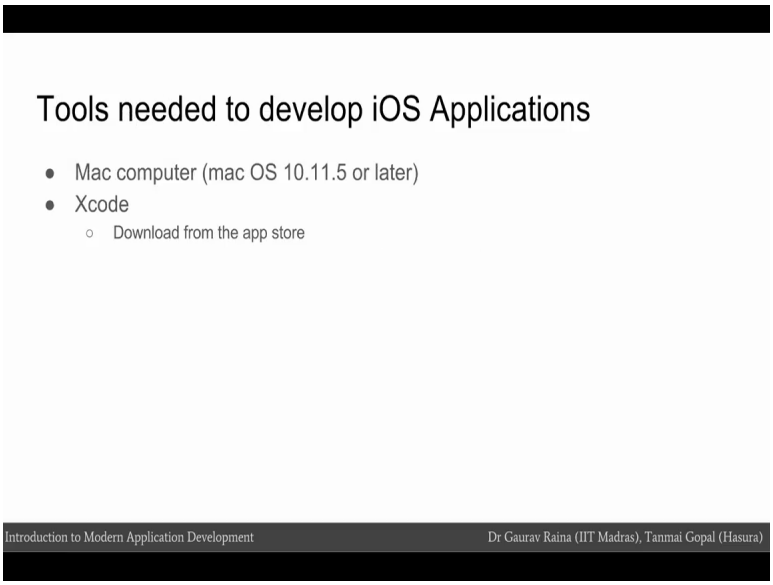
At any given moment your app is in one of the solving execution stage which is not running inactive active background or suspended. The system moves your app from state to state in responses to action happening throughout the system for example, when the user presses the home button or if a phone call comes in the currently running apps change a straighten response when the app is not been launched or if it is say terminated by the system it is in the not running state. In active state is when the app is running in the program, but is currently not receiving events and as usually stays in this state only briefly as it transitions to a different state.

When the app is in the active state it is running in the program and is receiving events this is the normal modes for foreground apps. When the apps is in the background and executing code it is said to be in the back state most apps enter this state briefly on their way to be suspended; however, an app that request extra execution time in a remain in this state for a period of time. In addition an app being launched directly into the background enters the state instead of the in active state. Finally, we have the suspended state in which the app is in the background, but it is not executing code the system moves the app to this state automatically and does not notifying them before doing. So, while suspended an app remains in memory, but does not execute any code when a low memory condition occurs the system may purge a suspended apps without notice to make more space for foreground app.

(Refer Slide Time: 06:22)

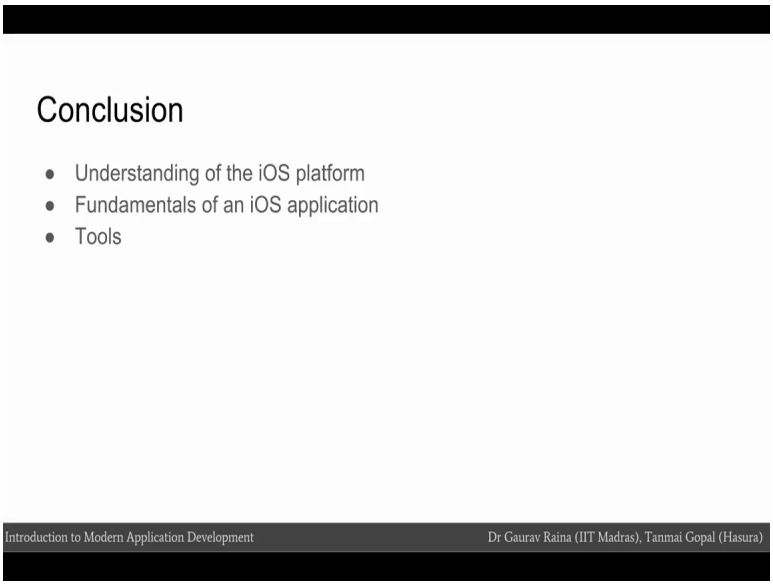
The App Delegate, the app delegate creates the window layer apps content is drawn and provides a place to respond to state transitions within the app. The app delegate is rare the entry point your app and the run loop is created this work is done by the UI application main attribute which is accessed in the app delegate.

(Refer Slide Time: 06:40)



To develop iOS apps using the latest technologies you need Mac computer which is running on Mac wise 10.11.5 or later and run in the latest version C code; X code includes all the features you need to design develop and debug an app. X code also includes the iOS decay which extends X code to includes the tools compiler frameworks and all the other things that you need specifically for iOS development you can download the latest version of X code on your Mac from the app store.

(Refer Slide Time: 07:18)



The X code only runs on a Mac hence it is not possible to develop iOS applications on any or anything by a Mac computer.

With this we come to the end of this module through this module we got an better understanding of the iOS platform. We have looked it specific iOS applications fundamentals and add allocate the tools required to develop iOS apps.