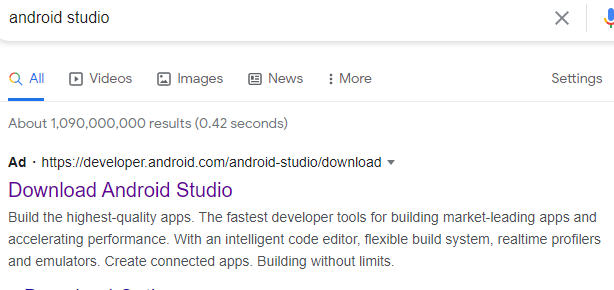
Subject: Mobile Computing  
Instructor: Prof. Haq Nawaz

**Week#1**

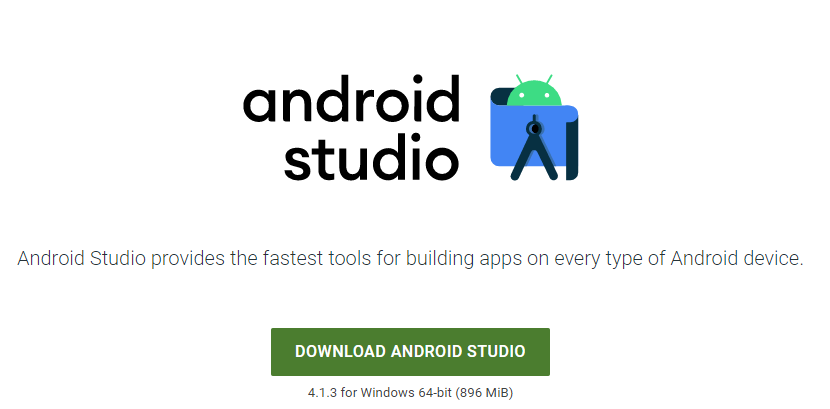
Lecture#1(February 23,2021)

Download Android Studio:

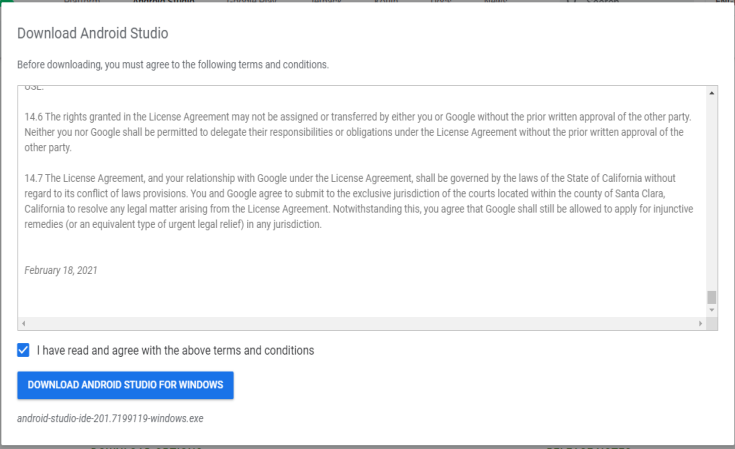
1. type **Android Studio**  on Google



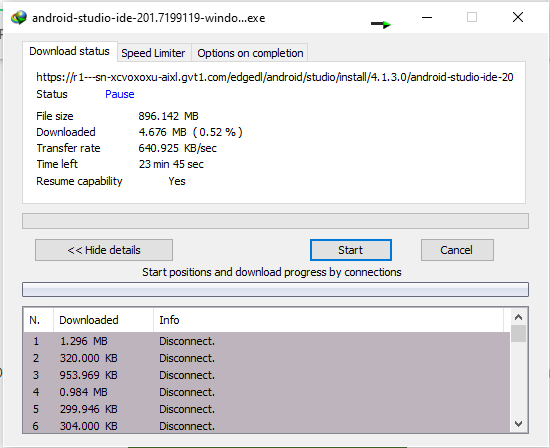
Click on **Download Android Studio**

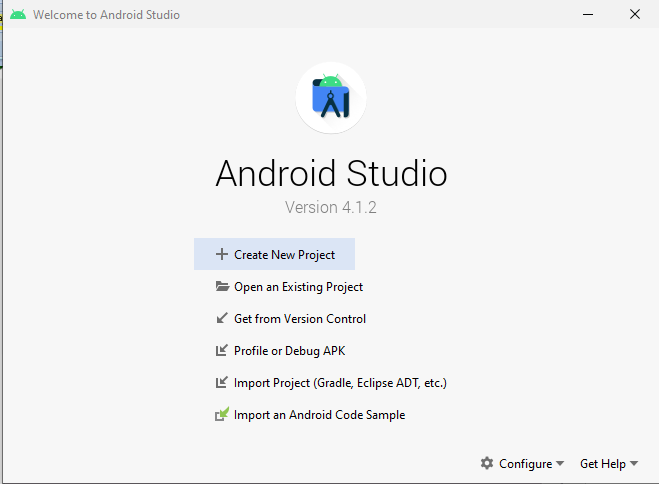
1. Now The Following Screen Appear 

Click on **DOWNLOAD ANDROID STUDIO**

1. Read all the terms and conditions

Click the check box and Click on **DOWNLOAD ANDRIOD STUDIO FOR WINDOWS**

1. Now Downloading is **Start**. Wait until the downloading will complete **.  
   **  
   After Downloading Install the **ANDRIOD STUDIO**
2. After Successful Installation of **Android Studio** You can create your project

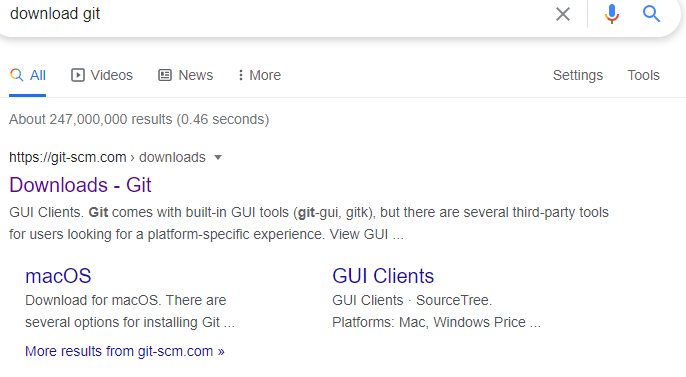
****

**End of Lecture#01**

Lecture#2( February 25,2021)

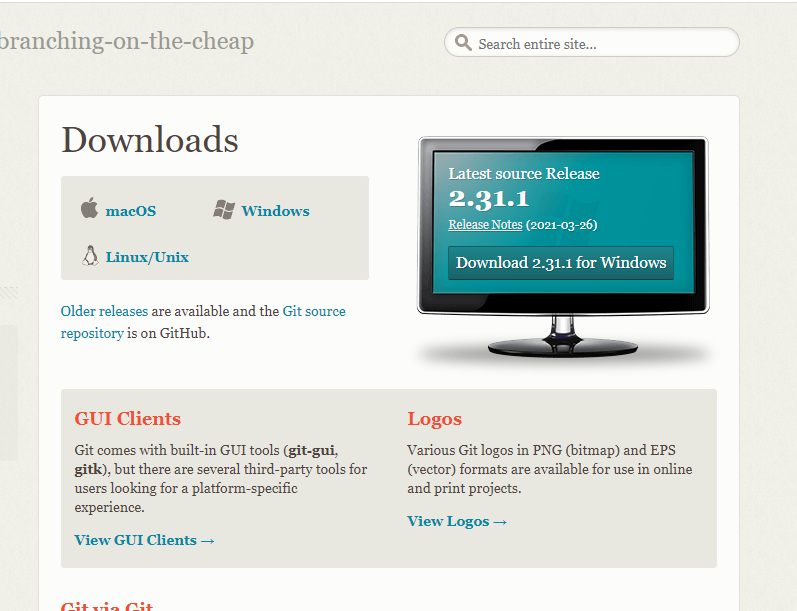
**Github**

1. To create github account first you have to download git
   1. From Browser type **Download git** / <https://git-scm.com/downloads>

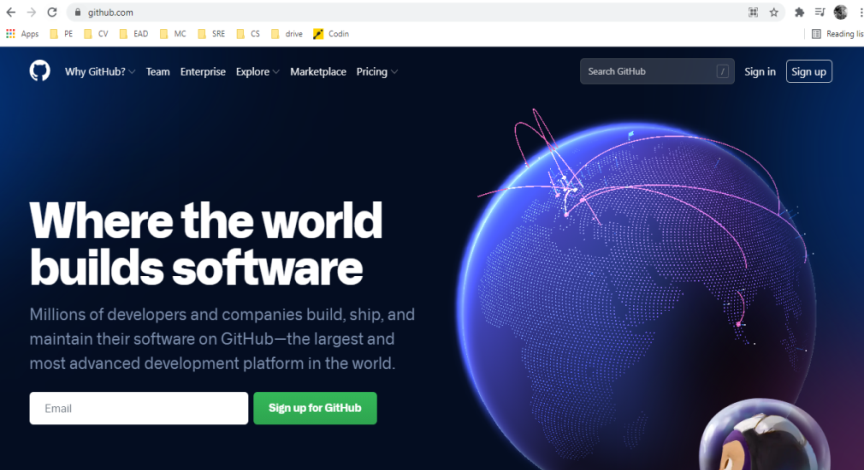


Click on **Downloads – Git**

1.2> The Following Screen will appear

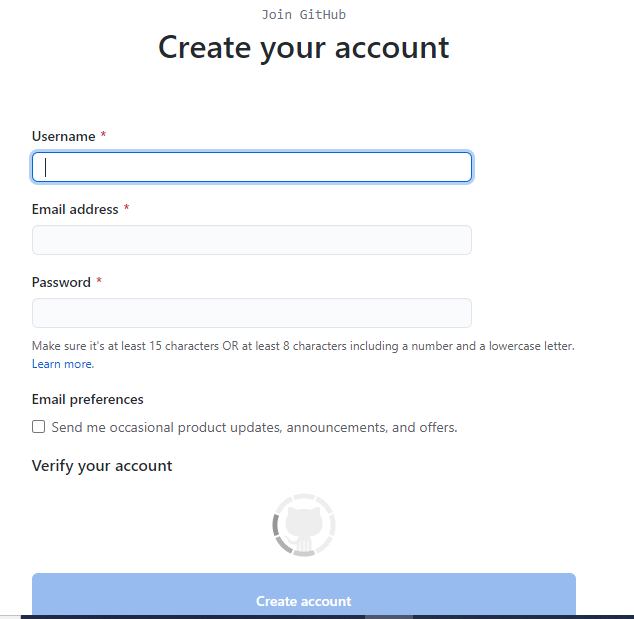
 Click on **Download 2.3.1 for Windows**  
 1.3> Wait until the downloading will complete than install it  
Now your git is install

2) **Github Account Creation**  
 2.1> From Browser type **github.com** or click <https://github.com/>



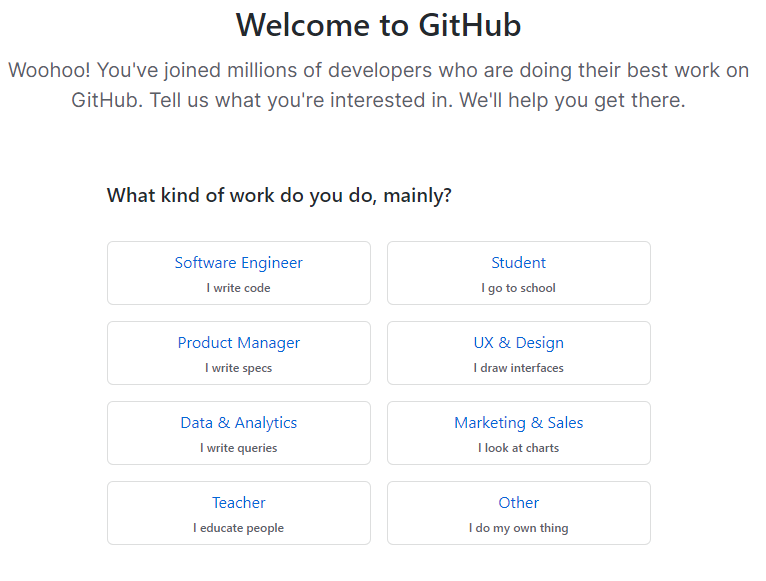
The following screen will appear. Now click on **Sign up for GitHub**

2.2> The following Form will appear



Fill the form and click on **Create Account** option

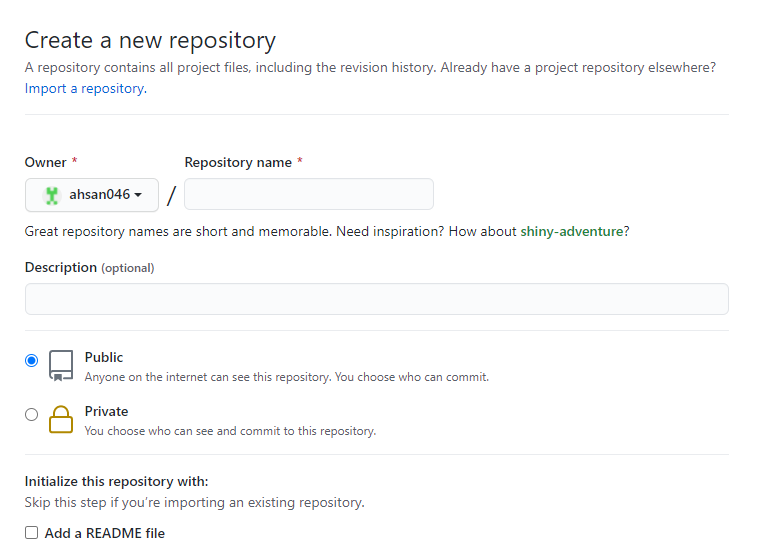
2.3> After account creation. The following screen will appear



***Now your Github Account has been successfully created***

3)  **Creating a new Repository on github website**

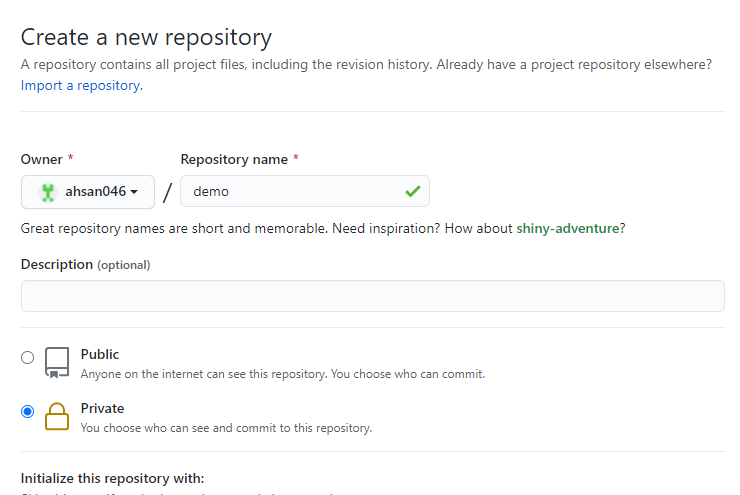
3.1> From Browser type **github.com/new** or click <https://github.com/new>



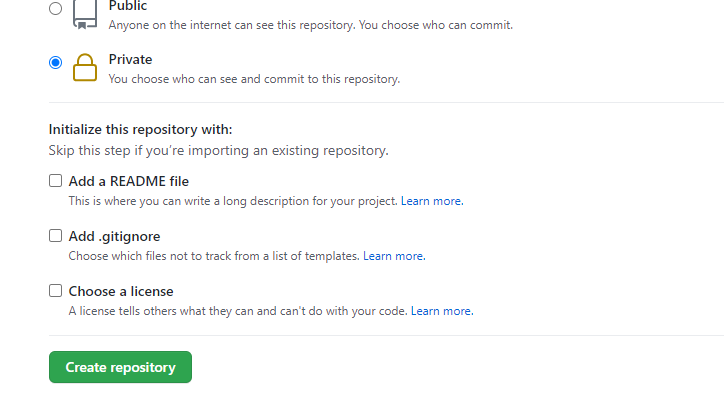
This will appear.

3.2> Now enter your repository name and make it public/ private on your

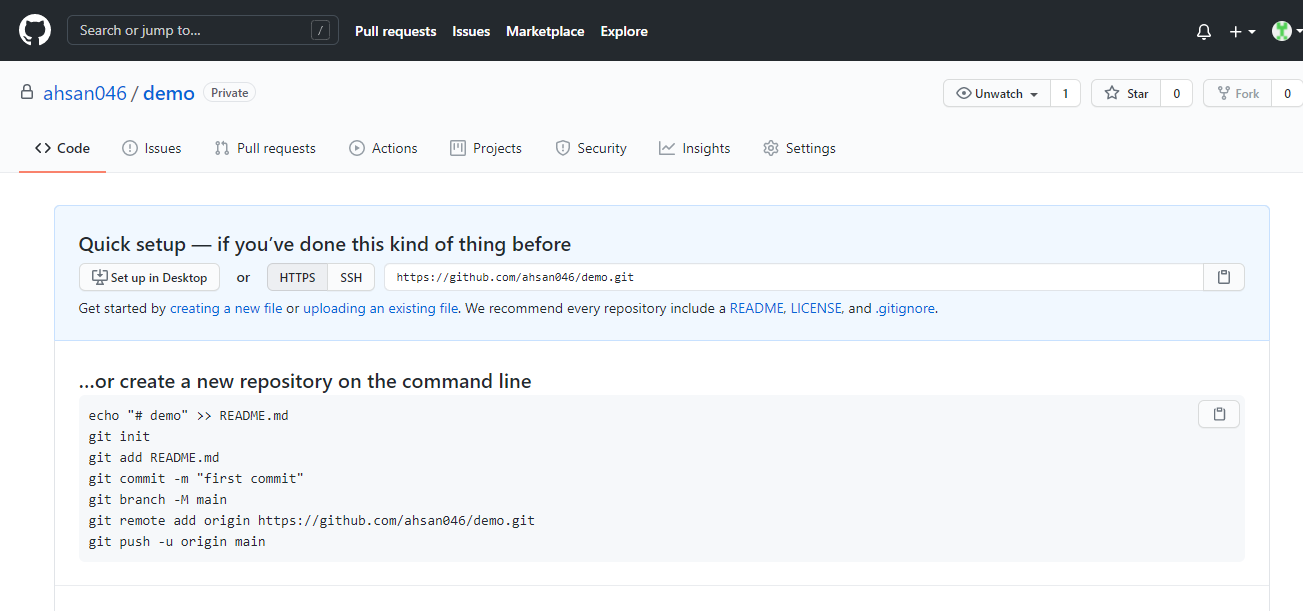
choice



3.3> Scroll down and click on **Create repository**

****

3.4> Now after repository creation the following screen will appear



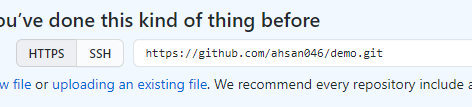
**Your Repository has been created**

1. After creating a repository online. Let's try to clone it, so that it can be

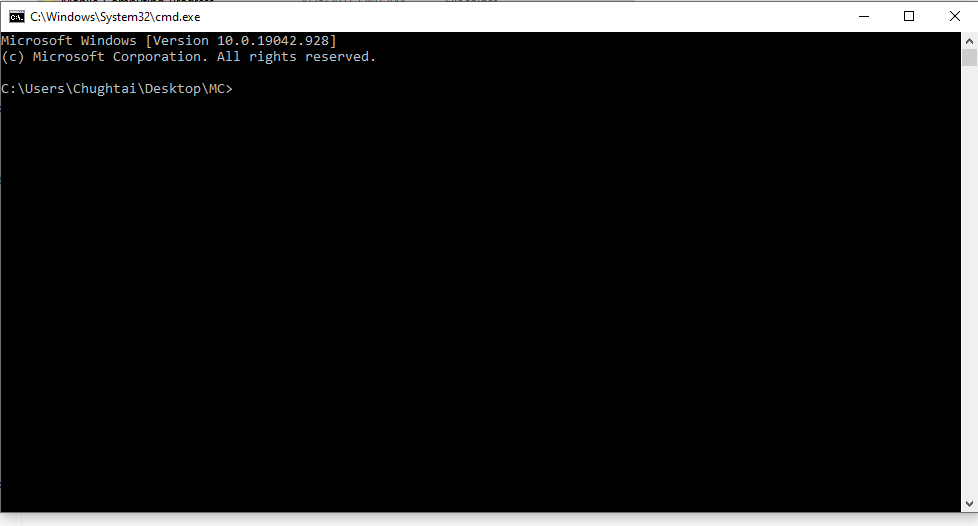
used on your PC.

4.1> Copy the repository URL ,you can either copy **HTTPS** or **SSH**. I will copy

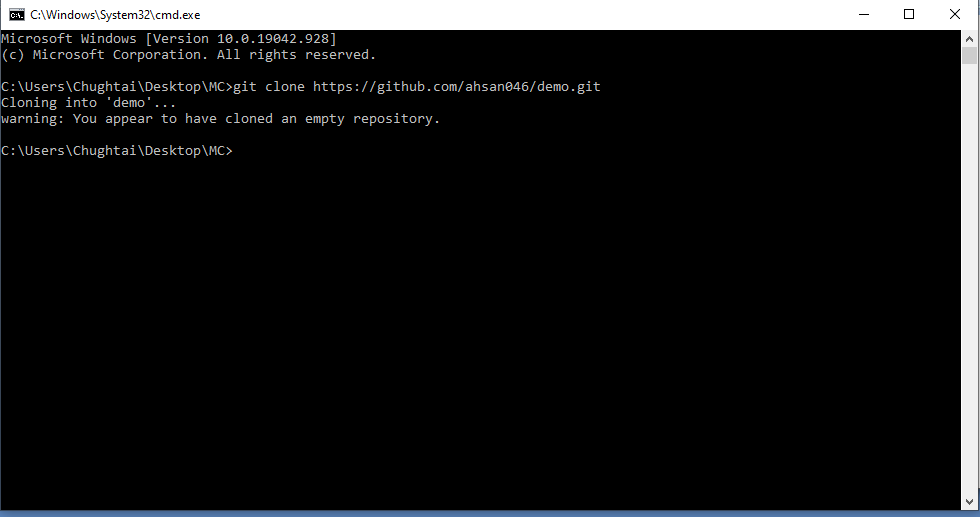
**HTTPS** URL.

****

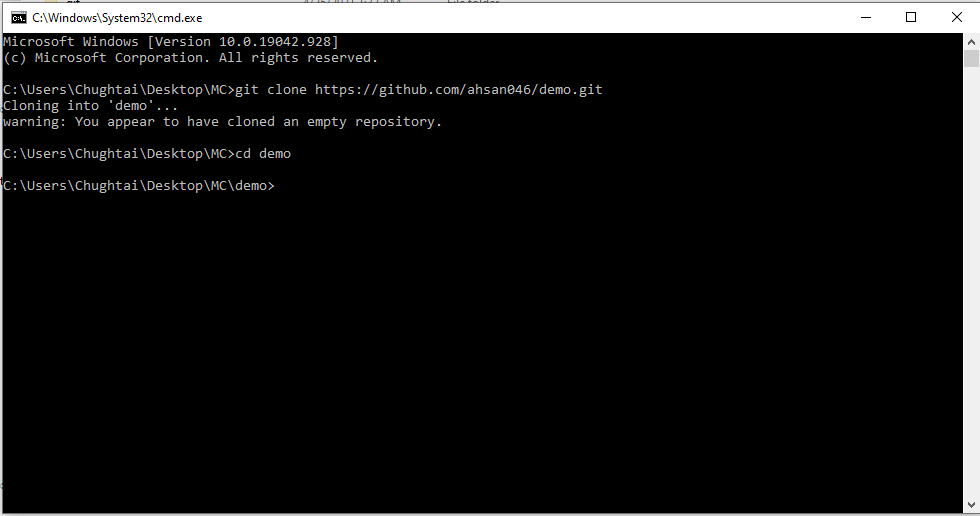
4.2> Go to your directory where you want to clone it and than open CMD

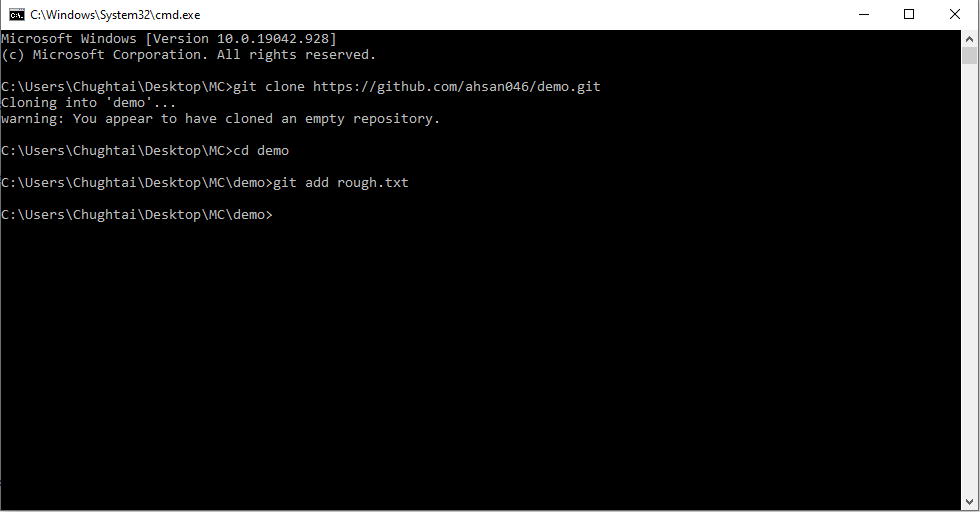


4.3> Now type **git clone** [**https://github.com/ahsan046/demo.git**](https://github.com/ahsan046/demo.git)

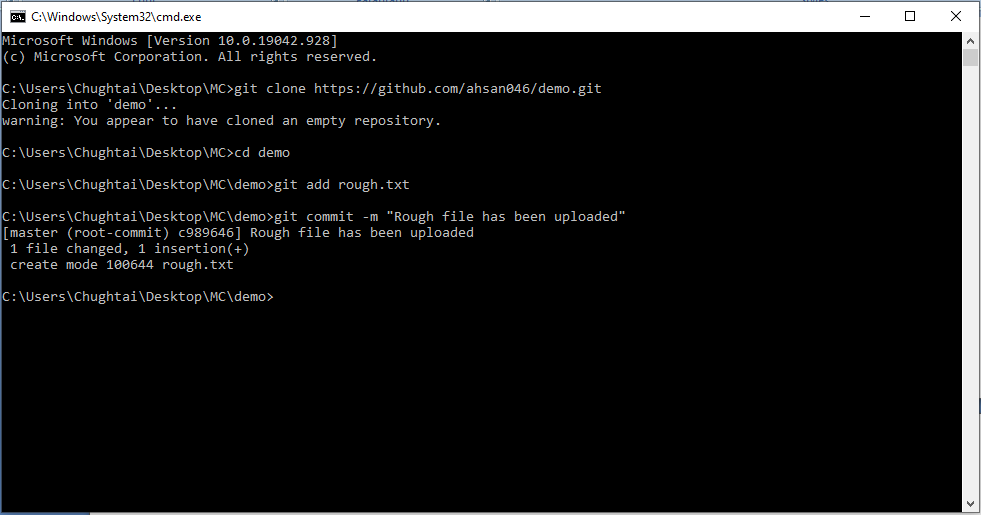


4.4> We have cloned the repository. Now add a new file and push it  
 First we have to go to that directory now type **cd directory\_name** or  
 you can go to that directory and again open the CMD

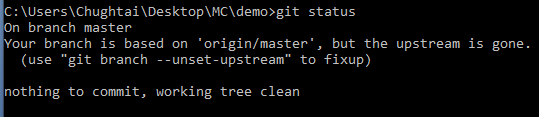


4.5> Create a new file which you want to push add something in it  
 Use the **git add FileName** Command enter the file that you created   
 

4.6> After adding we use git commit command with some message  
 Syntax **git commit –m “Message”**

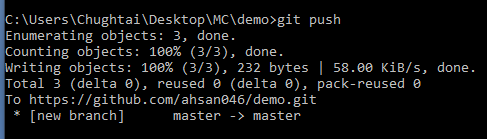


4.7> **git status** command is use to check the status of your repository



4.8>   Now we committed the changes. But these changes are not visible

online. The reason is that we have to push the repo. So enter **git** **push**

command. Now you can see changes on **Github repo** also.  
 

4.9> The **git pull** command is used to **fetch** and download content from a

remote repository and immediately update the local repository to

match that content.



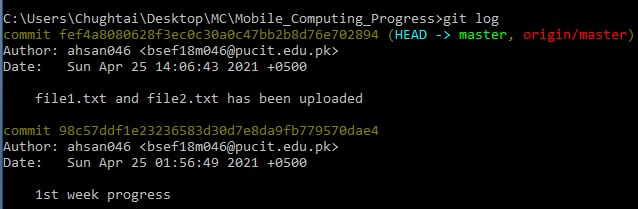
**End of Lecture#2**

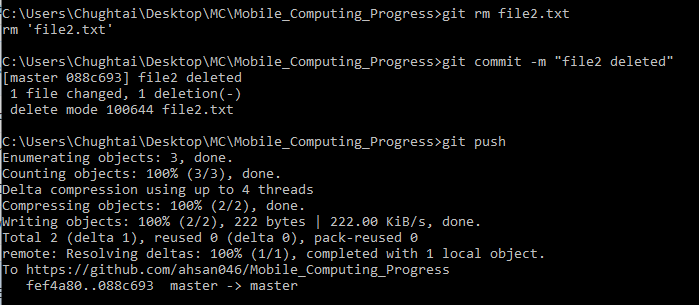
**End of Week#1**

**Week#2**

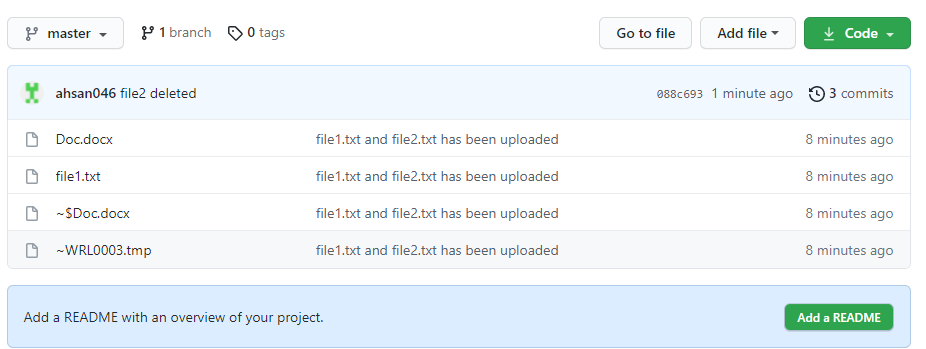
Lecture#3( March 02,2021)

**Git log:** tool to review and read history of everything in repository.

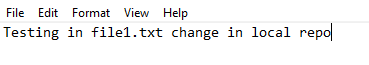
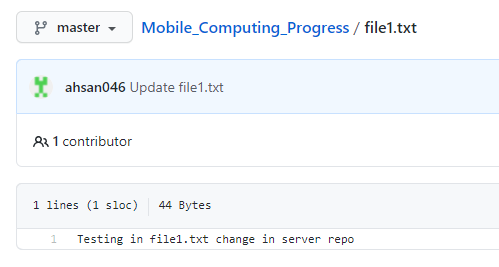
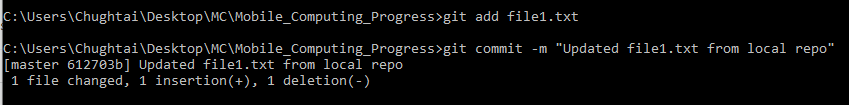
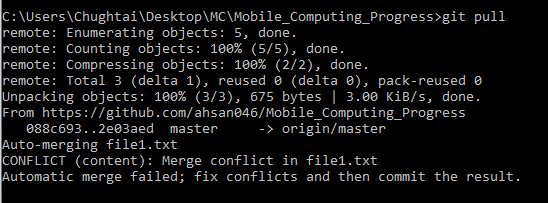


**git rm** command is used to delete the file  


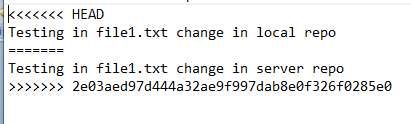
On pushing **git rm** command your file will delete from server repository



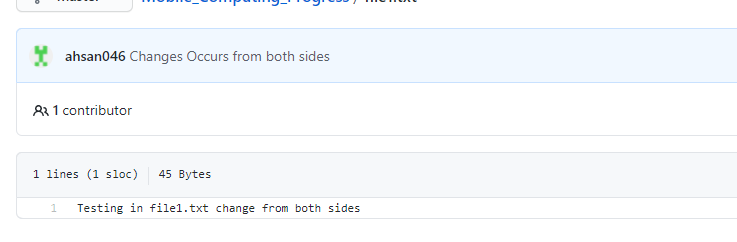
**Merge Conflict** :

Occurs when same thing change from 2 different place at one time  
 1) First we change at our local repo  
   
 2) Than we change same thing in github repo  
   
 3) Now we try to push our local file file1.txt  
   
 4) Now first we pull file1.txt  
 

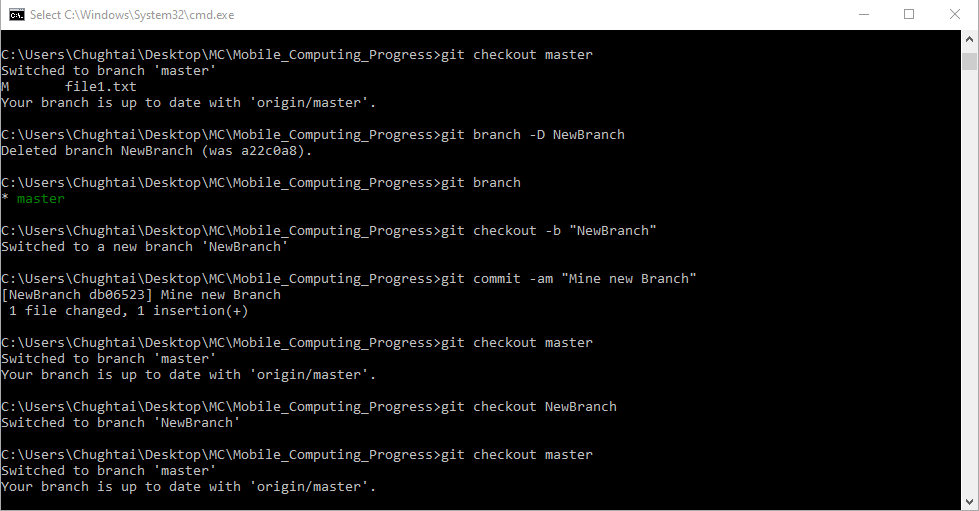
This will be appear on text editor

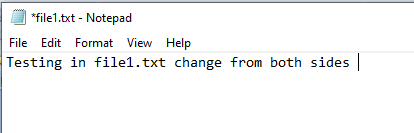


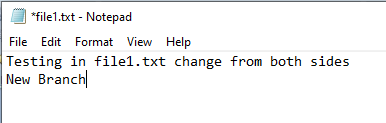
5) Now the issue is resolve . We can push it now

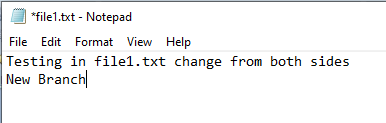


**GIT BRANCH**

GIT BRANCH command lets you create, list, rename, and delete branches. It doesn't let you switch between branches or put a forked history back together again. The git checkout – b “name” command is used to create new branch. Git branch show the list of branch.git checkout name command is used to change the branch. Git merge is used to merge branches.git branch –D name is used to delete the branch.  


Mine Master Branch File  
  
Mine NewBranch File

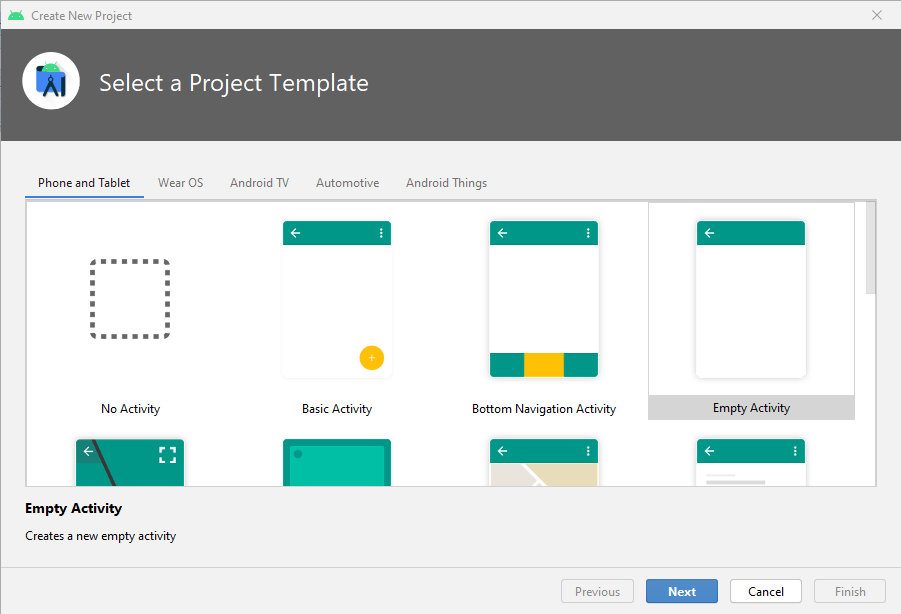
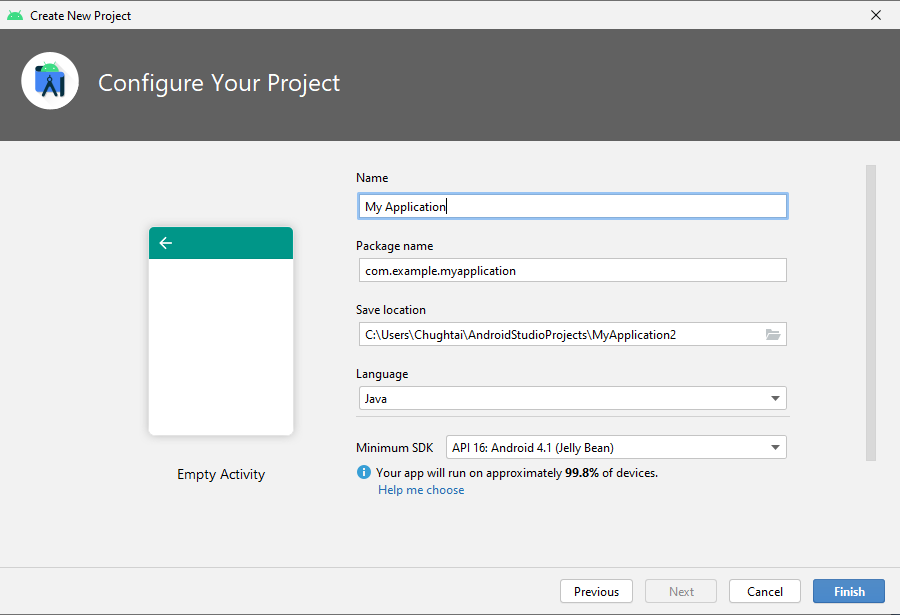
  
After merge the NewBranch in Master Branch



**End of Lecture#3**

Lecture#4(04 March, 2021)

**Android Studio Project Creation**

1. From Search bar type **Android Studio**   
      
    Select the **Empty Activity** and click on **Next** Button  
   2) The Following Page will appear  
    

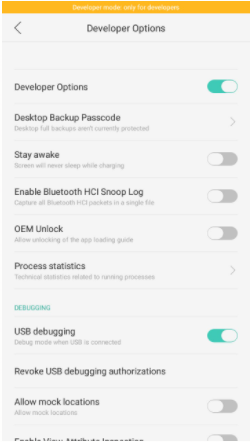
Simply give the name of your project and choose the **java** language

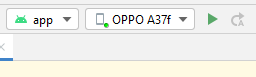
and click on **Finish** button you can also change the location of your

project from **Save Location**.

**Mobile Connectivity:**

1. Attach your mobile to laptop/PC through data cable
2. Go to Mobile Setting and on the **Developer Options**  than

**USB debugging** option  


1. After on of developer options and USB debugging option the Name of your device will show on Android Studio   
     
   Now you can run your project Application on your mobile.

**Android Studio:**

* .xml file is for Screen front end and java file for coding
* There is already built in 1 xml file name **activity­­\_main.xml**
* **activity­­\_main.xml** is the default screen
* In xml file there are 2 options

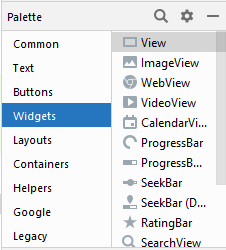
1. Code : You built your screen by typing
2. Design : You built you screen by drag and drop

**Android Studio Screen:**

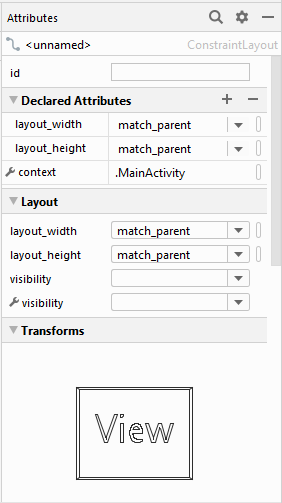
Now we will create 3 different Screen on Android Studio

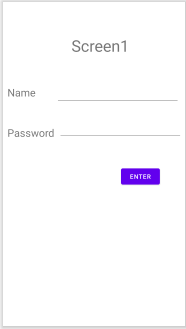
1. I will create Screen1 by Design tab

From Palette tab you can choose your required view/widget

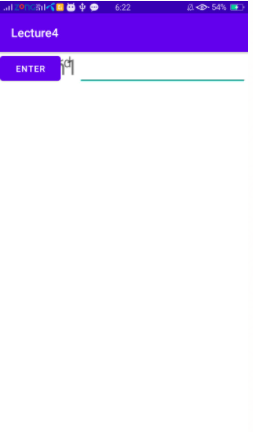


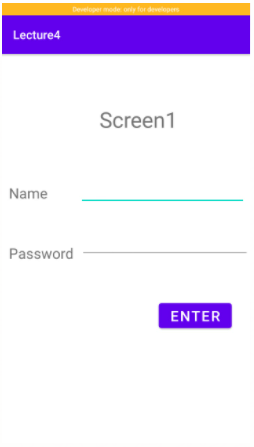
From Attribute tab you can set the attribute of your selected



* Without Constraint our Screen on Android Studio will show good like this  
  

But on running on mobile /Emulator all will set to top left of the Screen



* To solve this problem we will set Constraint to each View and we must select atleast their 1 x and 1 y axis constraint  
    
  Now it is working fine

1. I will create Screen2 from **Code** tab  
    The Output of my Screen2 is  
       
    Now explain the code of 1 TextView

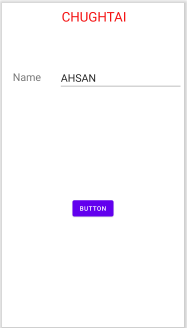
**Id:** A unique name for the element , which you can use to obtain a reference to view from your application

**Layout:** It defines the structure for a user interface in your app, such as in an activity. All elements in the **layout** are built using a hierarchy of View and ViewGroup objects. A View usually draws something the user can see and interact with.

**Margin and padding:** It help to arrange layout elements. Using them you could easily set gaps between elements, shift some elements and make more space around elements content.

**Constraint:** Itallow you to position a given widget relative to another one. You can **constrain** a widget on the horizontal and vertical axis: Horizontal Axis: left, right, start and end sides.

1. Screen3 is



**End of Lecture#4**

**End of Week#2**

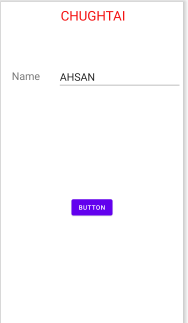
**Week#3**

Lecture#5(09 March,2021)

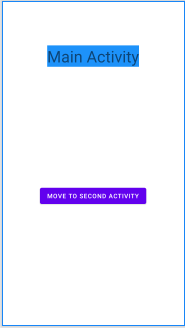
1. **Constraint Layout:** It simplifies creating complex **layouts** in Android by making it possible to build most of your UI using only the visual editor in Android Studio. **Layout** Editor in Android Studio.

The example of Constraint Layouts are

i)

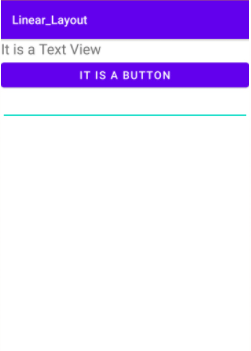


ii)



1. **Linear Layout:** It is the most basic **layout**, and it arranges its elements sequentially, either horizontally or vertically. To arrange controls within a **linear layout**

Followings are the output of LinearLayout



ii)



**Points Regarding Constraints:**

1. We must have to give the 1 x axis constraint and 1 y axis constraint
2. If we not give any top constraint the widget will move towards bottom
3. Similarly If we not give left constraint the widget will move towards right
4. If we not not give any constraint than the widget will set [0,0]

**End of Lecture#5**

Lecture#6(11 March ,2021)

Intents and Actions

**Intent:**

* It is a description of an operation to be performed.
* It is an object used to request an action from another app component via Android System
* An **Intent** can do

1. Start an Activity: Clicking share opens an app that allows you to post a photo
2. Start Service: Fetching data in background
3. Deliver Broadcast: Informs everybody that the phone is on charging

* **Types:**

There are 2 types of Intent

1. **Explicit Intent:**

Starts a specific Activity.

1. **Implicit Intent:**

Asks system to find an Activity that can handle this request.

**Activity:**

* It is an application component.
* It represents one window, one hierarchy of views.
* It typically fills the Screen, but can be embedded in other Activity or appear as floating window.
* It can start other activities in the same or can be in other apps.
* It handles user interfaces.
* Java class which typically has one Activity in one file.

**Implementation**

* I will create an Application having 2 activities in it

1. **Main Activity**

* It is the default activity name as Main Activity
* Having 1 **TextView** and 1 **Button**
* The TextView have the text of Main Activity
* The **Button** has the text of **Go** **to** **Second Activity**
* By clicking on the Button I call the **Intent** to Second Activity

1. **Second Activity**

* It is the second activity name as **Activity Second**
* Having 1 **TextView** and 3 **Button**
* The TextView have the text of Second Activity
* **Button1** has the text of **Go to Main Activity**
* By clicking on the Button1 I call the **Intent** to Main Activity
* **Button2** has the text of **Call**
* By clicking on the Call Button I call the intent of Call Log

By which I can call to the number that I pass to activity

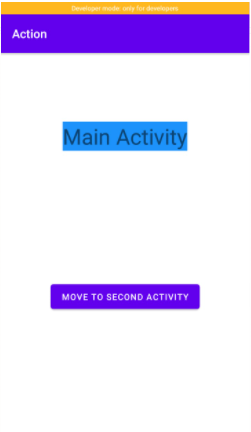
* **Button2** has the text of **WebSite**
* By clicking on the Call Button I call the intent of website

By clicking I will go to the website which URL I give in intent.

Now I Show you the Screen Shots of my Action Application

1. **MAIN ACTIVITY**

The default Activity



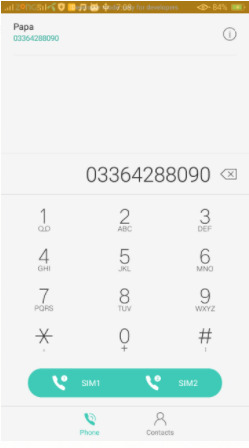
1. **SECOND ACTIVITY**

On clicking the Go To Second Activity we will on the Second Activity.



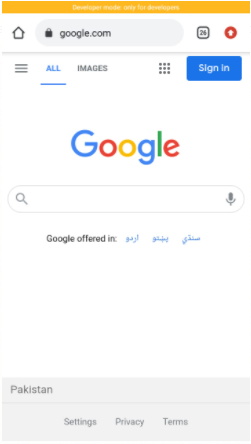
1. **CALL**

On clicking the **Call** Button in Second Activity



1. **WEBSITE**

On Clicking the **Website** Button in Second Activity



**End of lecture#6**

**End of Week#3**

**Week#4**

No Lecture on (16 March, 2021)

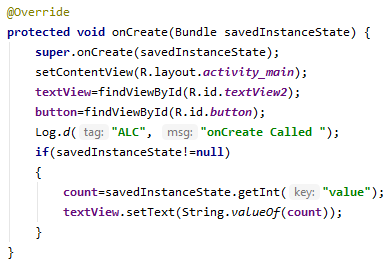
Lecture#7(18 March, 2021)

Activity Life Cycle

* To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of **6** callbacks
* The **6**  Callbacks are given below

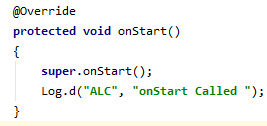
1. OnCreate()

Called for those activities which created with the attribute Persistable Mode set to PersistAcrossReboots.



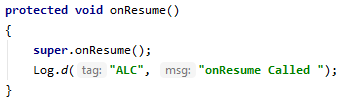
1. OnStart()

When activity start getting visible to user then **onStart**() will be called.It calls after onCreate() at first time launch of activity.



1. onResume()

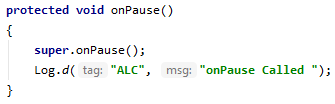
When the activity enters the Resume state, it comes to the foreground,& then the system invokes the onResume() callback.



1. onPause()

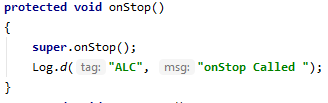
When Activity is in background than onPause() method will

Execute. At this the activity is not visible to user and goes in background when this method is executed



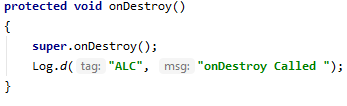
1. onStop()

After a millisecond of onPause() method next onStop() method will execute and this activity is aloso not visible to user when onStop() method will execute.



1. onDestroy()

  It a **method** called by the framework when your activity is closing down. It is called to allow your activity to do any shut-down operations it may wish to do

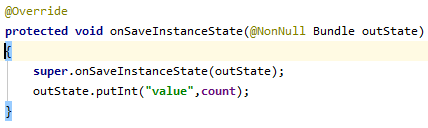


Problem in Activity

* When you call another activity from current activity or we rotate our mobile screen than the current Activity will go to **onPause()** state. When we come back from another activity to previous activity than it will be **onStart()** but the previous safed valued will be lost.

Solution

* We use **onSaveInstanceState(Bundle bundle)** to safe the value So we can reused them when our activity comes back from another activity.

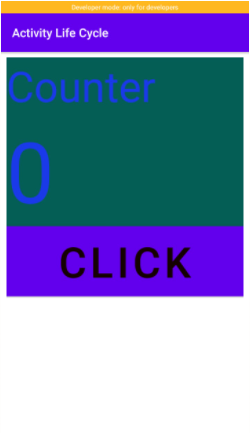


Application Explanation

Now I will give the ScreenShots of my Apllication and explain it

**When Activity Start at 1st time:**

When you open the application first time the following screen will display

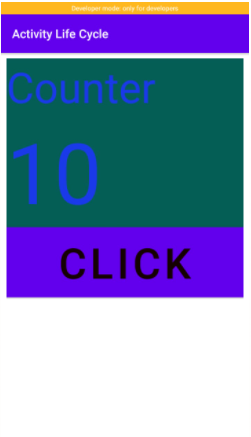


On first I have a textView having the value of Counter.

Second I have the textView having the counter value on first start its value will be zero.

On last I have a button on which click the value of the counter value will be increase by **1**.

After clicking 10 times on the Button



**On Screen Rotation**

On Screen rotation the Current activity go to onPause() state and I save the value by the use of **onSaveInstanceSave(Bundle)** and by rotaion I come back to this Activity onCreate() & I get my saved value from **onSaveInstanceSave (Bundle)** So my value will not set to zero by rotation.



**End of Lecture#7**

**End of Week#4**

**Week#5**

Lecture#8(25 March 2021)

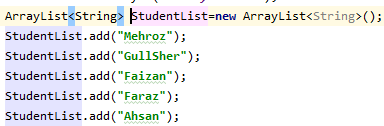
ListViews

Android Adapters

It is bridge between **UI components** & **data source** that helps us to fill data in UI component. It holds the data and send the data to an **Adapter** view then view can takes the data from the adapter view and shows the data on different views like ListView,GridView etc.

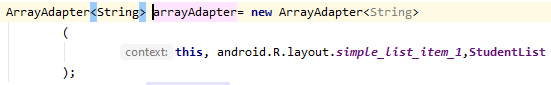
ArrayList

  It is a dynamic data structure in which you can add or remove any number of elements and those elements are stored in ordered sequence. It may also contain duplicate values.



ArrayAdapter

It holds the data & send the data to adapter view then view can takes the data from the adapter view & shows the data on different view like listview, gridview etc.



ListView Binding

It is a control that is used for display a list of data items. It inherits directly from ListBox. Listview provides an ItemsSource property for binding list of items.



Update of Record

**arrayAdapter**.notifyDataSetChanged()function is used to update the record



Sorting of Record

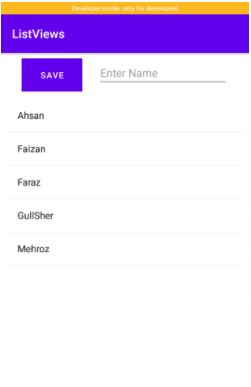
Collections.sort() function is used to sorting the record



Application Explanation

**When Appliaction will Start 1st time:**

When Application will start 1st time the following screen will display



The following Screen having 3 widgets.

* **Button**

Having the context of Save

By clicking on it the context of the EditText which is right on it will be save in listView

* **EditText**

On the riht of Save button EditText is used to to get the string which the user want to safe

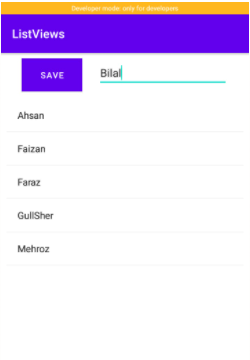
* **ListView**

It contains a list of string

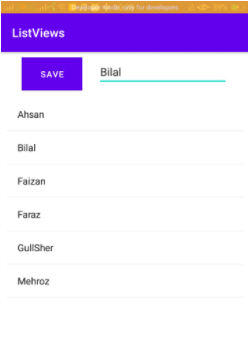
On clicking on any listview another activity will show having the context of that string on which you clicked

**Adding New Record:**

If you want to store a record in list than Enter a String which you want to store

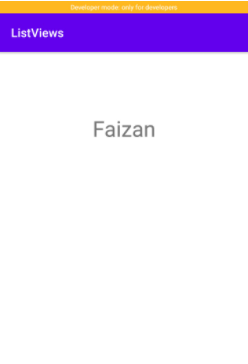


After Entering the string Just click on save Button. I enter **Bilal** an click on save button.



**On Clicking of Record:**

On clicking in listview the particular String Another activity will display having the context on which I clicked on **Faizan**



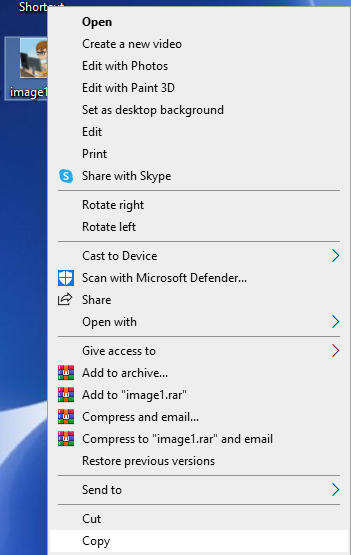
**End of Lecture#8**

Lecture#9(26 March 2021) MakeUp

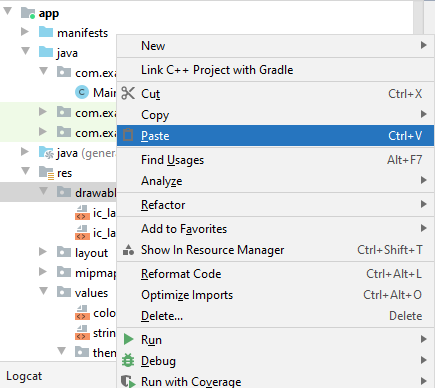
Image in Android Studio

Adding Image in Android Studio:

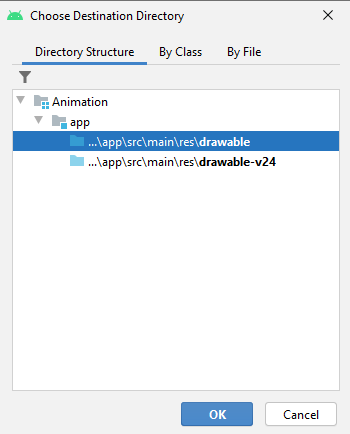
1. To add the image in android studio right click on that image which you want to add & copy it



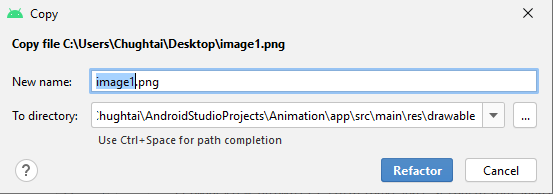
1. Now come to Android Studio right click on **drawable** and click on **Paste** button



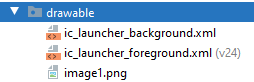
The following screen will appear just click on **OK** Button



After **OK** the following screen will appear   
From here you can give the name of your image   
After naming click on **Refactor** Button

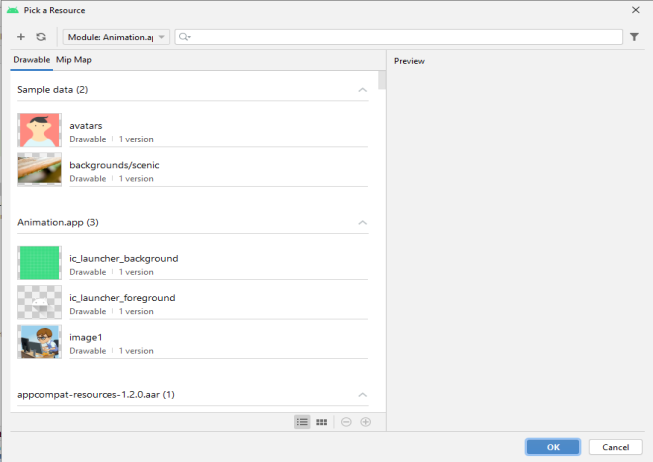


1. Now Image has been successfully safe in Android Studio & you can see from **drawable**



Using Image in Application:

1. Go to Your design tab of xml file and drag an **ImageView** after it the following screen will appear



Just select your desire Image and click on **OK** Button

You can also set the constraint and height as well as width of the image

Image Animation:

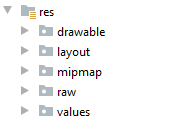
After dragging the image you can do many animations with images from java

Some of the animation I will explain here

* imageView.animate() is used for the animation of image
* setDuration(time in ms) is used to set the time of the following operation.
* Alpha(0 to 1) is used to control the alpha level of object
* By translate function you can shift the image in right,left,up down by giving the value
* Rotation is used to rotate the image in x and y direction. 360 means the 1 complete cycle and 180 means half rotation
* You can use more than 1 animation at 1 time

Video in Android Studio

1. Go to xml file and simply drag an videoView and set its height,width and set constraints.
2. Now copy the video which you want to add and paste it in raw folder of resource



1. Now create a **VideoView** variable and set its path



1. Create a **MediaController** variable and set its **AncherView** to the the video

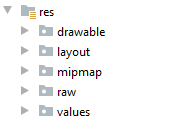


1. Now just give the mediaController variable to setMediaController of video and then just start the video

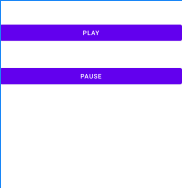


Voice in Android Studio

1. Copy the Voice which you want to add and paste it in raw folder of resourse file



1. Go to xml file and create 2 buttons **Play** & **Pause**



1. In java file just create MediaPlayer variable and give it the location of voice in raw folder



1. Now just in Play Function play the voice & in Pause Button pause the voice



**End of Lecture#9**

**End of Week#5**

**Week#6**

Lecture#10(30 March,2021)

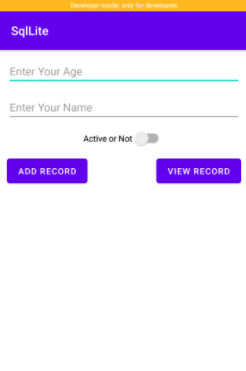
Sqlite

* It is an open-source relational database.
* It is used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database.
* **It**  is embedded in **android** by default. So, there is no **need** to perform any **database** setup or administration task.

Application Explanation

**When Application will Start First Time:**

When Application will **Start** first time. The following **Screen** will show



It contains

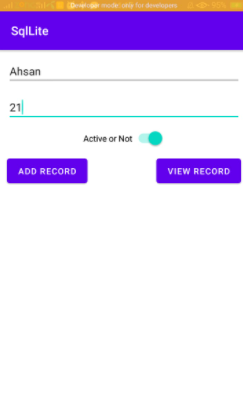
* On first It contains **EditText** having the hint of **Enter Your Name** the user enter his name

and in **back ends** it will safe in class of **CustomerModel** having type of **String** on clicking on **ADD RECORD**

* On Second It contains **EditText** having the hint of **Enter Your Age** the user enter his age
* and in **back ends** it will safe in class of **CustomerModel** having type of **Int** on clicking on **ADD RECORD**
* On third it contains **SwitchButton** having the value of **Active or Not**
* and in **back ends** it will safe in class of **CustomerModel** having type of **booleam** on clicking on **ADD RECORD**
* A button having value of **ADD RECORD** on clicking on the Toast will show having the data that you enter above.
* A button having value of **VIEW RECORD**

**Entering of data:**

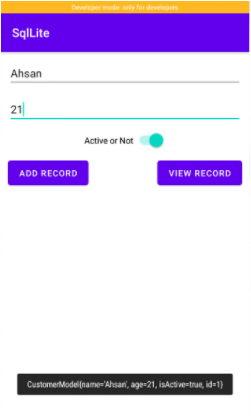
* The user can Enter his name in **EditText** of **Enter Your Name** in **String**
* The user can Enter his age in **EditText** of **Enter Your Age** in Int
* The user can Seelect whether he is Active or not by **SwitchButton**.



* I enter here my name , age & on the SwitchButton

**On ADD RECORD**

* On clicking the **ADD RECORD** button I used the function of **toString** & show the enter data in toast



* My class also has a id of type integer which is auto generated.

**End of lecture#10**

Lecture#11( 01 April , 2021)

Seekbar:

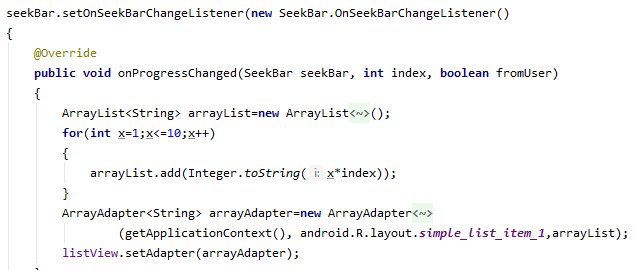
* It is an extension of ProgressBar that adds a draggable thumb.
* The user can touch thumb/finger and drag left or right to set the current level or use the arrow keys.
* Placing focusable widgets to the left or right of a Seekbar is discouraged.

Seekbar.max() Function is used to set the maximum value of the seekbar

For Example if you give the max value of the seekbar 100 than the value range is from 0 to 99 like an array which starts from 0



Seekbar.clickonchangeListner function is used to get the value of the seekbar every time when its changed by using this function of seekbar



As you see that in the function I am making a table of 1 to 10 of the valus which we get by index parameter and than shows all of its values in the list view

End of Lecture#11

**End of week#6**

**Week#7**

Lecture#12(06 April, 2021)

Drawer Navigation:

**End of Lecture#12**

Lecture#13(08 April,2021)

Only Evolution of AssignmentOne

**End of Lecture#13**

**End of Week#7**

**Week#8**

Lecture#14(13 April, 2021)

Dialog:

* It is a small window that prompts to make a decision or enter additional information
* A dialog does not fill the screen and is normally used for modal events that require users to take an action before they can proceed.

1. Dialog by **without New layout**

AlertDialog.builder :

Some times in your application, if you wanted to ask the user about taking a decision between yes or no in response of any particular action taken by the user, by remaining in the same activity and without changing the screen, you can use Alert Dialog.

In order to make an alert dialog, you need to make an object of AlertDialog.Builder which an inner class of Alert.Dialog. Its syntax is given below



Builder.setTitle() :

It is used to set the title of our dialog box



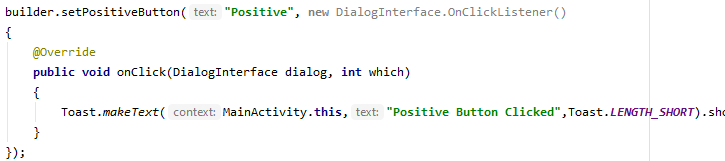
Bulider.setMessage():

It is used to give the message to the dialog



Builder.setPositiveButton():

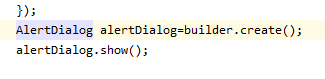
Set a listener to be invoked when the positive button of the dialog is pressed.



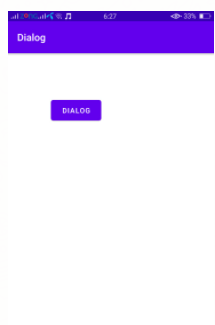
By clicking on the positive button the toast having the text of **Positive Button Clicked** will show.

AlertDialog.Show()

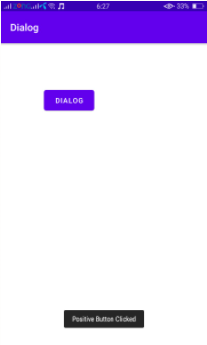
It is used to show the dialog box



**Application:**







**End of Lecture#14**