public void ClickFunction(View view) // UI element referring button is view  
 { //EditText is a data type , nameEditText is a variable  
 EditText nameEditText = (EditText)findViewById(R.id.*nameEditText*);  
// where nameEditText is the id of the text field , R is resources , id - to define set of resources  
 // the findViewById() function returns the graphical element  
  
 Log.*i*("info","You Pressed The Button!");  
 Log.*i*("value",nameEditText.getText().toString()); // here we are logging a value which we get from the text field with id = 'nameEditText'  
 }

public void onClickFunc(View view) //passsing the graphical UI element button  
{  
 EditText nameText = (EditText)findViewById(R.id.*nameText*);  
 EditText passwordText =(EditText)findViewById(R.id.*passwordText*);  
  
 Log.*i*("info", "You Have Successfully Logged in With The Following Information");  
 Log.*i*("Username :",nameText.getText().toString());  
 Log.*i*("Password :" ,passwordText.getText().toString());  
  
 Toast.*makeText*(this, "Hi There ! ", Toast.*LENGTH\_SHORT*).show();  
 //this refers to the app , text : to show what text should be displayed  
 // LENGTH\_SHORT OR LONG : how long message is displayed and show() : to show the Toast  
 // Message pops up from the bottom of the screen - that is why it is called toast  
  
  
}

Type Toast and click on the makeText function (not android widget) and the ide automatically fills the function parameters.

//for changing images where test.image is the id of the image being used.

ImageView image = (ImageView) findViewById(R.id.test\_image);

image.setImageResource(R.drawable.xxx);

**for switching between two images: simultaneously on the click of a button**

public void Switcher(View view) //passing the UI element button of type view  
{ i++; //button has been pressed  
 Log.*i*("Info", "Button Has Been Pressed "+ i +" Times");  
 ImageView img = (ImageView) findViewById(R.id.*iphone1*);  
 if(i%2==0) {  
  
 img.setImageResource(R.drawable.*iphone2*); //setImageResource function sets the new image  
 }  
 else  
 {  
 img.setImageResource(R.drawable.*iphone1*);  
 }  
}

public void convertcurrency(View view)  
{  
 Log.*i*("Info","Button Pressed ! ");  
 EditText currencyField = (EditText)findViewById(R.id.*currencyField*);  
  
 String amountINR = currencyField.getText().toString();  
 double INR = Double.*parseDouble*(amountINR); //for converting string to integer  
 double DHS = INR / 20;  
 //String DHString = Double.toString(DHS);  
 String DHString = String.*format*("%.2f",DHS); //converts to a string upto 2 decimal places  
 Toast.*makeText*(this, INR + " INR Is " + DHString + " Dirhams.", Toast.*LENGTH\_LONG*).show();  
  
  
 Log.*i*("Rupees", currencyField.getText().toString());  
}

**Making a class : and the respective logic for TriangularSqNoChecker –**

class Number  
{  
 public int num;  
  
 public boolean isTriangular()  
 {  
 int x=0; //counter  
  
 int tno=0;  
  
 while (tno<num)  
 {  
 tno=tno+x+1;  
 x++;  
 //tno is a triangular number , we keep on calculating tno till we surpass it.  
 }  
  
 if(tno==num)  
 return true;  
 else  
 return false;  
 }  
  
 public boolean isSquare()  
 {  
 int sqroot = (int) Math.*sqrt*(num); //also can use Math.floor(sqroot) == sqroot  
 if(sqroot\*sqroot == num)  
 return true;  
 else  
 return false;  
 }  
  
}  
  
public void testNumber(View view)  
{  
 Log.*i*("info","Button Pressed");  
  
  
 EditText numberField = (EditText) findViewById(R.id.*numberField*);  
 String message ;  
  
 if(numberField.getText().toString().isEmpty())  
 message = "Field Is Empty , Please Type In A Number & Try Again";  
  
 else {  
 Number a = new Number(); // a is an object of class Number  
 a.num = Integer.*parseInt*(numberField.getText().toString()); //converting number from string to int  
 if(a.num<0)  
 message = "Number is -Ve , Try Again";  
 else if (a.isSquare() && a.isTriangular())  
 message = a.num + " is A Triangular And Square Number !";  
 else if (a.isTriangular())  
 message = a.num + " is a Triangular Number But Not A Square Number";  
 else if (a.isSquare())  
 message = a.num + " Is A Square Number But Not A Triangular Number";  
 else  
 message = "It is neither a Triangular or a Square Number ";  
 }  
  
 Toast.*makeText*(this, message, Toast.*LENGTH\_SHORT*).show();  
}

Animations: fading :

public int i=1;  
  
 public void fade(View view)  
 {  
 Log.*i*("Info","Image Clicked");  
 i++;  
 ImageView img1 = (ImageView) findViewById(R.id.*img1*);  
 ImageView img2 = (ImageView) findViewById(R.id.*img2*);  
 if(i%2==0) {  
 //alpha refers to opacity of picture to 0 - fade out completely  
 img1.animate().alpha(0).setDuration(2000); //fading over 2 seconds  
 // we set alpha of homer img to 0 after copy pasting xml code so it is not visible intially , then we switch alpha to 1.  
 // alpha has values between 1 and 0  
 img2.animate().alpha(1).setDuration(2000);  
 }  
 else  
 {  
 img1.animate().alpha(1).setDuration(2000);  
 img2.animate().alpha(0).setDuration(2000);img2.animate().alpha(1).setDuration(2000);  
 }  
 }

To animate bart upon opening of the app : put animate function in onCreate class (App Opening):

//THIS MAKES BART FLY IN FROM THE LEFT , WE INITIALLY SET BART ON THE LEFT AND BRING HIM BACK TO ORIGINAL POSITION BY ROTATION AND TRANSLATION.

@Override  
protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 ImageView img1 = (ImageView) findViewById(R.id.*img1*); //img1 is bart  
 //instant setting x without animations  
 img1.setX(-1000); //we set bart image off screen  
 //Aim: bart should come spinning on , and at first he is faded , he should come spinning from the left  
 //remove constraints from design  
 img1.animate().translationX(1).rotation(3600).setDuration(3000);  
}

**//Game Connect :**

We use a grid view and then in each of the grid view , we place the tokens and adjust the margins to accomadate the image of the yellow/red token – we do this once for one of the elements and copy paste the setting for all the other elements , after which, we remove the token image, we create an onclick function , where we set the imageview(only the structure of the token) outside the screen and within the onclick , we set the new image depending upon the active player =0 or 1 and we animate the imageview or token using translation to pull it down, we also set tags to each of the counter. see code below.

/we use margins to fix the margins of the tokens w.r.t grid , check xml code , all margin:10dp , left margin:15dp  
 //also we put the grid using grid view and put rows and columns and fixed the background img using attributes  
 //and we copy pasted one image view onto every row and column in xml cod

public class MainActivity extends AppCompatActivity {  
//we use margins to fix the margins of the tokens w.r.t grid , check xml code , all margin:10dp , left margin:15dp  
 //also we put the grid using grid view and put rows and columns and fixed the background img using attributes  
 //and we copy pasted one image view onto every row and column in xml code  
  
 //0:yellow , 1:red , 2: Empty - Initially all elements in the game state are 2 : empty  
 int activePlayer =0;  
 int[] gameState = {2,2,2,2,2,2,2,2,2}; // representing the color or no element in each counter space  
  
 int winningPositons [][] = {{0,1,2},{3,4,5},{6,7,8},{0,4,8},{2,4,6},{0,3,6},{1,4,7},{2,5,8}}; //horizontally , vertically or diagonally  
  
  
 boolean gameActive = true;  
 public void dropIn(View view) //click function on counter spaces  
  
 {  
 ImageView counter = (ImageView) view; //image is initially null. we reference the view here directly - not getting any id as it is a common function  
  
 Log.*i*("INFO", counter.getTag().toString()); //gets tag of each element 0,1 to 8 whenever that element is clicked  
  
  
 int tag = Integer.*parseInt*(counter.getTag().toString()); //we get the tag number  
 if (gameState[tag]==2 && gameActive == true ) //we will only allow to insert counter if there is not counter present , or counter is initially empty && if the game is active  
  
 {  
 gameState[tag] = activePlayer; //setting the game state of the token on the basis of the activePlayer , where 2 = no active player  
  
  
 counter.setTranslationY(-1500); // setting the imageview outside the screen  
 if (activePlayer == 0) {  
 counter.setImageResource(R.drawable.*yellow*); //setting new image with yellow counter on click  
 activePlayer = 1; //red's chance next;  
 } else {  
 counter.setImageResource(R.drawable.*red*); //setting new image with yellow counter on click  
 activePlayer = 0; //goes back to yellow  
 }  
 counter.animate().translationYBy(1500).rotation(3600).setDuration(300); //pulling the image down by animation  
  
  
 for (int[] i : winningPositons) //we are looping through each element- check notebook  
 {  
 if (gameState[i[0]] == gameState[i[1]] && gameState[i[1]] == gameState[i[2]] && gameState[i[0]] != 2) {  
 //Someone Won  
 gameActive = false;  
 String message;  
 if (activePlayer == 1)  
 message = " Yellow Has Won !";  
 else  
 message = "Red has Won";  
  
 Button Retry = (Button)findViewById(R.id.*Retry*);  
 TextView WinnerMessage = (TextView) findViewById(R.id.*WinnerMessage*);  
 WinnerMessage.setText(message);  
 Retry.setVisibility(View.*VISIBLE*);  
 WinnerMessage.setVisibility(View.*VISIBLE*);  
  
 }  
 }  
 //No winner  
 int count=0;  
 for(int i:gameState)  
 {  
 if(i!=2)  
 count++;  
 }  
 if(count>8)  
 {  
 Button Retry = (Button)findViewById(R.id.*Retry*);  
 TextView WinnerMessage = (TextView) findViewById(R.id.*WinnerMessage*);  
 WinnerMessage.setText("No One Won");  
 Retry.setVisibility(View.*VISIBLE*);  
 WinnerMessage.setVisibility(View.*VISIBLE*);  
 }  
  
 }  
 }  
  
 public void PlayAgain(View view)  
 {  
 Button Retry = (Button) findViewById(R.id.*Retry*);  
 TextView WinnerMessage = (TextView) findViewById(R.id.*WinnerMessage*);  
 Retry.setVisibility(View.*INVISIBLE*);  
 WinnerMessage.setVisibility(View.*INVISIBLE*);  
 //making a variable of Grid Layout for looping through it's contents :  
 GridLayout gridLayout = (GridLayout) findViewById(R.id.*gridLayout*);  
 for(int i=0;i<gridLayout.getChildCount();i++)  
 {  
 ImageView counter = (ImageView) gridLayout.getChildAt(i); //we are getting the child contents of the gird layout instead of finding them by id having type image view at each ith iteration  
 counter.setImageDrawable(null);  
 }  
  
 for(int i=0;i<gameState.length; i++) //array length - length  
 {  
 gameState[i]= 2; //setting to empty  
 }  
 activePlayer =0;  
 gameActive = true;  
  
  
 }

**Videos:**

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 //we write functions on oncreate method if we want it to execute as soon as we open app:  
  
 VideoView videoView = (VideoView) findViewById(R.id.*videoView*);  
  
 videoView.setVideoPath("android.resource://" + getPackageName() + "/" + R.raw.*demovideo*); //setting video path  
  
 //creating an object called mediaController of class MediaController  
 MediaController mediaController = new MediaController(this);  
 //anchoring the mediaController to the video view  
 mediaController.setAnchorView(videoView);  
 //setting media controller for the video :  
 videoView.setMediaController(mediaController);  
 videoView.start();

}

**For Audio & Seeking Related Functions:**

public class MainActivity extends AppCompatActivity {  
  
MediaPlayer mediaPlayer;//creating a variable outside for referencing both functions  
AudioManager audioManager;  
 public void playAudio (View view)  
 {  
  
  
 mediaPlayer.start();  
 }  
  
 public void pause(View view)  
 {  
  
 mediaPlayer.pause();  
 }  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
//Setting up the media player during creation of the app , so it can pause&play the mediaplayer accordingly  
 mediaPlayer = MediaPlayer.*create*(this, R.raw.*marbles*);  
 //WE HAVE TO GET THE MAX VOLUME OF THE DEVICE:  
 //SETTING UP AUDIO MANAGER  
 audioManager = (AudioManager) getSystemService(*AUDIO\_SERVICE*);  
 int maxVolume = audioManager.getStreamMaxVolume(AudioManager.*STREAM\_MUSIC*);  
 final int curVolume = audioManager.getStreamVolume(AudioManager.*STREAM\_MUSIC*);  
 SeekBar volumeControl = (SeekBar)findViewById(R.id.*volumeSeekBar*);  
 volumeControl.setMax(maxVolume); //Setting max volume of the SEEKBAR  
 volumeControl.setProgress(curVolume); //Initially setting progress as current volume

volumeControl.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {

@Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 Log.*i*("Seekbar\_Changed",Integer.*toString*(progress));  
 audioManager.setStreamVolume(AudioManager.*STREAM\_MUSIC*,progress,0);  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 //SETTING UP SCRUB SEEKBAR  
  
 });  
  
 final SeekBar scrubSeekBar = (SeekBar) findViewById(R.id.*scrubSeekBar*);  
 //WE HAVE TO SET THE SEEK BAR MAX TO THE AUDIO DURATION IN THE BEGINNING :  
 scrubSeekBar.setMax(mediaPlayer.getDuration());  
 scrubSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 Log.*i*("ScrubSeek\_Changed",Integer.*toString*(progress));  
 //WHENEVER PROGRESS OF SEEK BAR IS CHANGED , we have to seek the audio  
 mediaPlayer.seekTo(progress);  
  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
 mediaPlayer.pause(); //pauses whenever user touches the bar  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
 int length = mediaPlayer.getCurrentPosition();  
 mediaPlayer.seekTo(length);  
 mediaPlayer.start(); //resumes at paused length when user stops touching the seekbar   
 }  
 });  
//regularly updating the progress of the seekbar as the audio goes by even without user interaction - we use Timer scheduled at a fixed rate  
 new Timer().scheduleAtFixedRate(new TimerTask() {  
 @Override  
 public void run() {  
 scrubSeekBar.setProgress(mediaPlayer.getCurrentPosition());  
 }  
 },0,400); //every 200 millisecond it is updated - scheduled at a fixed rate  
  
 }

**To access multiple audio files using one function by accessing the audio files (as a variable):**

public void playPhrase(View view)  
{ Button button =(Button) view; //we get all the buttons directly  
 Log.*i*("Button Pressed:", button.getTag().toString());  
 //if(button.getTag().toString() == "hello")  
  
 MediaPlayer m1 = MediaPlayer.*create*(this, getResources().getIdentifier(button.getTag().toString(), "raw", getPackageName()));  
 m1.start();  
 /\*tag name will be the same name as the audio file ,  
 so what we do here is get the tag name from the button pressed- PASS THAT THROUGH getIdentifier() and  
 check if there is an audio file with the same name and we use that to  
 create the media player whenever button is pressed  
 \*/  
}

**List view :**

protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 ListView myListView = (ListView) findViewById(R.id.*myListView*);  
  
 final ArrayList<String> Family = new ArrayList <String>();  
  
 Family.add("Bobby");  
 Family.add("Simi");  
 Family.add("Sidharth");  
 Family.add("Visakh");  
  
 //Connecting ListView and ArrayList - array adapter  
//initializing adapter with array  
 ArrayAdapter<String> arrayAdapter = new ArrayAdapter<String>(this,android.R.layout.*simple\_list\_item\_1*,Family);  
//Simple List Item 1 - BASIC LIST with LABELS  
 //connects the list view and adapter:  
 myListView.setAdapter(arrayAdapter);  
//function to define what happens on Clicking the item - that's  
 myListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
 @Override  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 //parent refers to the whole LIST  
 //view is the individual item clicked  
 //position of item in list  
 //id - similar to id - not required  
  
 Log.*i*("Name Clicked:", Family.get(position));  
 }  
 });  
 }

**App:Times Tables**

public class MainActivity extends AppCompatActivity {  
 ListView myListView;  
 public void generateTimesTable(int timesTableNo)  
 {  
 Log.*i*("SeekBar\_Changed",Integer.*toString*(timesTableNo));  
  
 final ArrayList <Integer> Numbers = new ArrayList <Integer>();  
  
 for(int i=1;i<=100;i++)  
 { Numbers.add(i\*timesTableNo);  
 }  
  
 myListView = (ListView)findViewById(R.id.*myListView*);  
 ArrayAdapter<Integer> arrayAdapter = new ArrayAdapter<Integer>(this , android.R.layout.*simple\_list\_item\_1*,Numbers);  
 myListView.setAdapter(arrayAdapter);  
 }  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 final SeekBar seekBar = (SeekBar)findViewById(R.id.*seekBar*);  
 int max=20;  
 int startingPos=10;  
 seekBar.setMax(max);  
 seekBar.setProgress(startingPos);  
 generateTimesTable(startingPos);  
  
 seekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 int min =1;  
 int timesTableNo;  
 if(progress < min) {  
 timesTableNo = min; //if slider is at 0 , we put the timesTablesNo at 1  
 seekBar.setProgress(min);//setting the seekbar's progress at 1 if it comes to 0  
 }  
 else  
 timesTableNo = progress; //if not below 0 , timesTableNo will be at the progress  
 generateTimesTable(timesTableNo);  
  
 }  
  
 @Override  
 public void onStartTrackingTouch(SeekBar seekBar) {  
  
 }  
  
 @Override  
 public void onStopTrackingTouch(SeekBar seekBar) {  
  
 }  
 });  
  
 }

**Two Types of Timers:**

new CountDownTimer(10000,1000){  
 public void onTick(long millisecondsUntilDone){  
 Log.*i*("Seconds Left:",String.*valueOf*(millisecondsUntilDone/1000));  
 }  
  
 public void onFinish(){  
 Log.*i*("We're Done !","No More Countdown");  
 }  
}.start();  
  
  
  
final Handler handler = new Handler();  
  
//handler runs the runnable  
  
Runnable run = new Runnable() {  
 @Override  
 public void run() {  
 Log.*i*("hey , it's us", "A second passed by");  
  
 handler.postDelayed(this,1000);  
 }  
};  
handler.post(run);

**Making an EGG Timer app:**

public class MainActivity extends AppCompatActivity {  
 MediaPlayer mediaPlayer;  
 TextView timerTextView;  
 SeekBar timerSeekBar;  
 Boolean CounterIsActive = false;  
 Button goButton;  
 CountDownTimer countDownTimer;  
  
 public void resetTimer()  
 {  
 timerTextView.setText("0:30");  
 timerSeekBar.setProgress(30);  
 timerSeekBar.setEnabled(true);  
 countDownTimer.cancel();  
 goButton.setText("GO!");  
 CounterIsActive = false;  
 }  
  
 public void buttonClicked(View view)  
 { //For stopping the timer  
 if(CounterIsActive) //if counter is already active when button is clicked , this is the STOP condition  
 {  
 resetTimer();  
 }  
 else //if timer isn't active , this is executed - go condition  
 {  
 CounterIsActive = true;  
 timerSeekBar.setEnabled(false);  
 goButton.setText("STOP");  
 countDownTimer = new CountDownTimer(timerSeekBar.getProgress()\*1000+100, 1000)  
 {//timerSeekBar.getProgress gives the progress of seekbar in milliseconds.  
  
 @Override  
 public void onTick(long millisUntilFinished) {  
 updateTimer((int)millisUntilFinished/1000); //we change the textview and update timer once countdown starts  
 }  
  
 @Override  
 public void onFinish() {  
 Log.*i*("Finished","Timer Done!");  
 mediaPlayer.start();  
 resetTimer();  
  
 }  
 }.start();  
 }  
 }  
 public void updateTimer(int progress)  
 {  
 int minutes = progress/60;  
 int seconds = progress - (minutes\*60);  
 String minuteString = Integer.*toString*(minutes);  
 String secondString = Integer.*toString*(seconds);  
 if(seconds==0)  
 secondString = "00";  
 if(seconds<10)  
 secondString = "0" +secondString;  
 if(minutes<10)  
 minuteString = "0"+minuteString;  
  
 timerTextView.setText(minuteString + ":" +secondString);  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 mediaPlayer = MediaPlayer.*create*(this, R.raw.*airhorn*);  
 timerSeekBar = (SeekBar) findViewById(R.id.*timerSeekBar*);  
 goButton = (Button)findViewById(R.id.*goButton*);  
 timerTextView = (TextView) findViewById(R.id.*countdownTextView*);  
 timerSeekBar.setMax(600); //10 minutes - 60 seconds  
 timerSeekBar.setProgress(30);  
  
 timerSeekBar.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {  
 @Override  
 public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {  
 updateTimer(progress);  
  
 }

**Hiding & Showing UI Elements: setVisibility(…)**

public class MainActivity extends AppCompatActivity {  
TextView textView ;  
 public void show(View view){  
  
 textView.setVisibility(View.*VISIBLE*);  
 }  
 public void hide(View view)  
 {  
 textView.setVisibility(View.*INVISIBLE*);}  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 textView = (TextView)findViewById(R.id.*textView*); //create the textview on the oncreate method:  
 }

* In grid layout, try hardcoding with xml , don’t forget to include app:layout:columnWeight and rowWeight = 1 and layout:column and row (no) as well as android:tag = “”

**Brain Teaser Program:**

package visakh.myappcompany.brainteaser20;  
  
import android.os.CountDownTimer;  
import android.support.constraint.ConstraintLayout;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.lang.reflect.Array;  
import java.util.ArrayList;  
import java.util.Random;  
  
public class MainActivity extends AppCompatActivity {  
  
 Button goButton;  
 int locationOfCorrectAnswer;  
 ArrayList <Integer> answers = new ArrayList <Integer>(); //Consists of one correct answer and three wrong answers  
 TextView resultTextView;  
 TextView scoreTextView;  
 TextView sumTextView;  
 TextView timerTextView;  
 int score=0;  
 int numberOfQuestions =0;  
 Button button0,button1,button2,button3;  
 Button playAgainButton;  
 ConstraintLayout gameLayout;  
  
  
 public void chooseAnswer(View view)  
 { if(Integer.*toString*(locationOfCorrectAnswer).equals(view.getTag().toString())) // we use view.getTag() - returns button the user pressed  
 {  
 resultTextView.setText("Correct !");  
 score++;  
 }  
 else  
 resultTextView.setText("Wrong !");  
 numberOfQuestions++; // Number of Questions Attempted Increases Irrelevant of correct/wrong answer  
 scoreTextView.setText(Integer.*toString*(score) + "/" + Integer.*toString*(numberOfQuestions)); //setting scoreTextView  
 newQuestion(); //after someone has chosen the correct answer - we need to generate a newQuestion  
  
 }  
  
 public void start(View view)  
 {  
 goButton.setVisibility(View.*INVISIBLE*);  
 playAgain(findViewById(R.id.*playAgainButton*)); //any view is fine as we are not using it - playAgain function just resets the Timer & starts the game  
 }  
  
 //Once Play Again Is Clicked :  
  
 public void playAgain(View view)  
 {  
 //Reset the game  
 gameLayout.setVisibility(View.*VISIBLE*);  
 score = 0;  
 numberOfQuestions = 0;  
 timerTextView.setText("30s");  
 scoreTextView.setText(Integer.*toString*(score) + "/" + Integer.*toString*(numberOfQuestions)); //update or reset the score counter  
 // Changing the Result Text View In The Beginning  
 resultTextView.setText("");  
 newQuestion();  
  
 playAgainButton.setVisibility(View.*INVISIBLE*);  
  
 //Creating a timer  
 new CountDownTimer(30100,1000)  
 {  
  
 @Override  
 public void onTick(long millisUntilFinished) {  
 timerTextView.setText(String.*valueOf*(millisUntilFinished/1000 + "s")); //for converting from long to string - String.valueOf(long) :  
  
 }  
  
 @Override  
 public void onFinish() {  
 resultTextView.setText("Done !");  
 playAgainButton.setVisibility(View.*VISIBLE*);  
 Toast.*makeText*(MainActivity.this, "You Got " + scoreTextView.getText().toString() , Toast.*LENGTH\_SHORT*).show();  
  
 }  
 }.start();  
  
 }  
  
 public void newQuestion()  
 {  
 //For Creating a Random Number Generation use class Random  
 Random rand = new Random();  
  
 int a = rand.nextInt(21);  
 int b = rand.nextInt(21);  
 // Setting Sum Text or Question:  
 sumTextView.setText(Integer.*toString*(a) + " + " + Integer.*toString*(b));  
  
 locationOfCorrectAnswer = rand.nextInt(4); //location of the correct answer can be anyone from 0-3 button  
  
 //Cleaning out the array of answers before adding in new answers  
 answers.clear();  
  
 for(int i=0; i<4; i++) {  
 if (i == locationOfCorrectAnswer)  
 answers.add(a + b); // i can be anywhere from 0-3 : so it could be assigned a random indices  
 else {  
 int wrongAnswer = rand.nextInt(41);  
 while (wrongAnswer == a + b) {  
 wrongAnswer = rand.nextInt(41); //if the wrong answer is same as the right answer , generate a new random wrong answer  
 }  
 answers.add(rand.nextInt(41));  
 }  
 } //End of for loop  
 //Since answers arrayList has 3 wrong answers and 1 correct answers , all of which are assigned at random  
 button0.setText(Integer.*toString*(answers.get(0)));  
 button1.setText(Integer.*toString*(answers.get(1)));  
 button2.setText(Integer.*toString*(answers.get(2)));  
 button3.setText(Integer.*toString*(answers.get(3)));  
 }  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 sumTextView = findViewById(R.id.*sumTextView*);  
  
 button0 = findViewById(R.id.*button0*);  
 button1 = findViewById(R.id.*button1*);  
 button2 = findViewById(R.id.*button2*);  
 button3 = findViewById(R.id.*button3*);  
 resultTextView = findViewById(R.id.*resultTextView*);  
 scoreTextView = findViewById(R.id.*scoreTextView*);  
 timerTextView = findViewById(R.id.*timerTextView*);  
 goButton = findViewById(R.id.*goButton*);  
 playAgainButton = findViewById(R.id.*playAgainButton*);  
 gameLayout = findViewById(R.id.*gameLayout*);  
  
 goButton.setVisibility(View.*VISIBLE*);  
 gameLayout.setVisibility(View.*INVISIBLE*);  
  
  
  
  
  
 }  
}

* Threads are basically where all your operations are running , main Threads – default thread is where all of your operations run by default. But if for example , if you want download html or web content or do web requests – then you have to use the background thread as the main Thread will be occupied with other user related operations , and if the main thread were used , those operations would come to a halt.
* // when we use ASyncTack - we can perform background operations quickly - so DownloadTask is now capable of performing tasks in the background  
  public class DownloadTask extends AsyncTask<String,Void,String>  
  {  
    
   @Override  
   protected String doInBackground(String... strings) {  
   return null;  
   }  
  }

**URL , HTML DOWNLOADING , etc :**

package visakh.myappcompany.downloadingwebcontent;  
  
import android.os.AsyncTask;  
import android.renderscript.ScriptGroup;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
  
import java.io.InputStream;  
import java.io.InputStreamReader;  
import java.net.HttpURLConnection;  
import java.net.MalformedURLException;  
import java.net.URL;  
  
public class MainActivity extends AppCompatActivity {  
  
 //Creating class to download html content  
 // when we use ASyncTack - we can perform background operations quickly - so DownloadTask is now capable of performing tasks in the background  
 public class DownloadTask extends AsyncTask<String,Void,String> //pass in a String(URL) - void - give out a String(HTML)  
 {  
 @Override //protected available throughout the whole package , String....strings - similar to arrays  
 protected String doInBackground(String... urls) {  
  
 String result = null;  
 URL url;  
 HttpURLConnection urlConnection = null; // used for grabbing the html from the url  
  
 //when you intialize URL , click on red alert and it implements try catch automatically  
 try {  
 url = new URL(urls[0]);  
 //open connection with url and type cast it to HttpURLConnection  
 urlConnection = (HttpURLConnection) url.openConnection();  
 InputStream in = urlConnection.getInputStream();  
 InputStreamReader reader = new InputStreamReader(in);  
 int data = reader.read();  
  
 while(data!= -1)  
 {  
 char current = (char) data;  
 result+=current; //letter by letter, it gets stored into the result string - which will consist of the html document  
 data = reader.read();  
 }  
  
 return result;  
  
 } catch (Exception e) {  
 e.printStackTrace();  
  
 return "Failed";  
 }  
  
 }  
 }  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 DownloadTask task = new DownloadTask();  
 String result = null;  
 try {  
 result = task.execute("https://zappycode.com").get(); // the .get gets the return value //executes basically adds whatever present in parameter and passes it strings[0] , we can add multiple URL's , like ,"www.google.com" - strings[1]  
 }  
 catch(Exception e)  
 {  
 e.printStackTrace(); //Logs the error  
 }  
 Log.*i*("Result ",result); //logs the html code   
  
  
 }  
}

**Steps for downloading Image from a URL :**

1. Create a class which extends AsyncTask<string,void,Bitmap>
2. Implement the function doInBackground with type Bitmap as we are returning a Bitmap
3. Within the try block – Declare a new Url and initialize with urls[0]
4. Then create an HTPP url connection and open connection with url decalred earlier – type cast to HttpUrlConnection
5. Connection.connect() – for connecting the HttpUrlConnection
6. Declare and InputStream var – in and getInputStream of connection
7. Then Declare a bitMap var which decodes stream of BitmapFactory with ‘in’ as parameter & then return the bitmap
8. <uses-permission android:name="android.permission.INTERNET"></uses-permission>

Don’t forget to get uses permission – Internet while using HTML files in androidmanifest.xml

* And for the image URL – click the image you want and “copy image address” – not image link.

**Application for dowloading images :**

package visakh.myappcompany.downloadingimages;  
  
import android.graphics.Bitmap;  
import android.graphics.BitmapFactory;  
import android.os.AsyncTask;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.util.Log;  
import android.view.View;  
import android.widget.ImageView;  
  
import java.io.IOException;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.MalformedURLException;  
import java.net.URL;  
  
public class MainActivity extends AppCompatActivity {  
  
 ImageView imageView;  
  
 public void downloadImage(View View)  
 {  
 ImageDownloader task = new ImageDownloader();  
 Bitmap myImage;  
 try {  
 myImage = task.execute("https://s3mn.mnimgs.com/img/shared/userimages/userimage\_16277929.jpg").get(); //returns the bitmap or the image  
  
 imageView.setImageBitmap(myImage);  
 }  
 catch(Exception e)  
 {  
 e.printStackTrace();  
 }  
  
  
 }  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 imageView = findViewById(R.id.imageView); //created image view on creation  
 }  
  
 public class ImageDownloader extends AsyncTask<String, Void , Bitmap> //ASyncTask is for background html downloading , we taking in a URL ---- and the we returning an image called Bitmap , we use ASyncTask so that html downloading is done in the background  
 {  
  
 @Override  
 protected Bitmap doInBackground(String... urls) {  
  
 try {  
 URL url = new URL(urls[0]);  
  
 HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
  
 connection.connect();  
  
 InputStream in = connection.getInputStream();  
  
 Bitmap myBitMap = BitmapFactory.decodeStream(in); //passing in the inputStream  
  
 return myBitMap; //returns that image  
  
 } catch (MalformedURLException e) {  
 e.printStackTrace();  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 catch(Exception e)  
 {  
 return null;  
 }  
 return null;  
 }  
 }  
}

**Java Program to split a String using Split:**

package domain;

import java.util.Arrays;

public class HelloWorld {

public static void main(String[] args) {

String myString ="Nick x Sean x Fido x Sara";

String [] splitString = myString.split(" x ");

System.out.println(Arrays.toString(splitString));

}

}

**Program to split a String using a subString:**

public static void main(String[] args) {

String river ="Mississippi";

String riverPart = river.substring(4,8); **//Doesn't include character at 8.**

System.out.println(riverPart);

}

**Java Program to Find Patterns In Strings :**

import java.util.regex.Pattern;

import java.util.regex.Matcher;

public class HelloWorld {

public static void main(String[] args) {

String river ="MississippiMississippiMississippi";

Pattern p = Pattern.compile("Mi(.\*?)pi"); // .\*? is the part between Mi and pi which is stored in p

Matcher m = p.matcher(river);

while(m.find()){ //keeps looking for the pattern in the String

System.out.println(m.group(1));

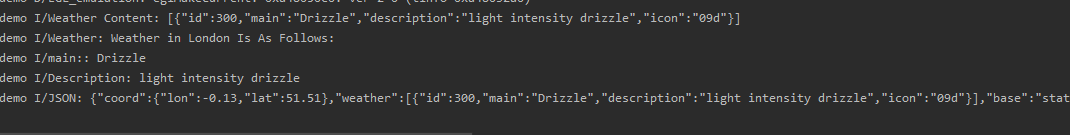
}

}

}

**JSON Program :**

Output:

****

**Code:**

public class MainActivity extends AppCompatActivity {  
  
 public class DownloadTask extends AsyncTask<String,Void,String>  
 {  
  
 @Override //doInBackground can't interact with the UI  
 protected String doInBackground(String... urls) {  
  
 String result = "";  
 URL url;  
 HttpURLConnection urlConnection = null;  
  
 try {  
 url = new URL(urls[0]);  
 urlConnection = (HttpURLConnection) url.openConnection();  
 InputStream in = urlConnection.getInputStream();  
 InputStreamReader reader = new InputStreamReader(in);  
 int data = reader.read();  
 while(data!=-1)  
 {  
 char current = (char) data;  
 result+=current;  
 data= reader.read();  
 }  
 return result;  
 }  
 catch(Exception e)  
 {  
 e.printStackTrace();  
 return "Failed";  
 }  
 }  
  
 @Override //onPostExecute can interact with UI  
 protected void onPostExecute(String s) {  
 super.onPostExecute(s);  
 //Processing JSON  
  
 try {  
 JSONObject jsonObject = new JSONObject(s);  
  
 String weatherInfo = jsonObject.getString("weather"); //key title : weather for getting all of it's weather content & returns an array of it's content ,I/Weather Content: [{"id":300,"main":"Drizzle","description":"light intensity drizzle","icon":"09d"}]  
  
 Log.*i*("Weather Content",weatherInfo);  
  
 JSONArray arr = new JSONArray(weatherInfo);  
  
 for (int i=0;i<arr.length();i++)  
 {  
 JSONObject jsonPart = arr.getJSONObject(i); //jsonPart parts the arr consisting of thw weather elements  
  
 Log.*i*("Weather","Weather in London Is As Follows:");  
 Log.*i*("main:",jsonPart.getString("main"));  
 Log.*i*("Description", jsonPart.getString("description"));  
 }  
 }  
 catch(Exception e)  
 {  
 e.printStackTrace();  
 }  
 Log.*i*("JSON ",s); //It Should log the JSON we put in task.execute()  
  
 }  
 }  
  
  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 DownloadTask task = new DownloadTask();  
 task.execute("https://samples.openweathermap.org/data/2.5/weather?q=London,uk&appid=b1b15e88fa797225412429c1c50c122a1");  
  
 }  
}