# EX.NO.1 ANDROID

Android is an open source and Linux-based **Operating System** for mobile devices such as smartphones and tablet computers. Android was developed by the *Open Handset Alliance*, led by Google, and other companies.

**A SOFTWARE DEVELOPMENT KIT**

* A software development kit (SDK or "devkit") is typically a set of software development tools that allows the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.
* To create applications you have to download this software development kit If you want to create an app
* Android app you require an SDK with java programming
* for iOS apps you require an iOS SDK with swift language.
* to develop MS Windows apps you require the .net language.
* There are also SDKs that are installed in apps to provide analytics and data about activity. Examples: Google and Face book

**INTERFACES(API)**

* Implementation of one or more application programming interfaces (APIs) in the form of some libraries to interface to a particular programming language or hardware that can communicate with a particular embedded system.
* Tools such as debugging facilities and other utilities, that are presented in an integrated development environment (IDE).
* A software engineer typically receives the SDK from the target system developer. SDK can be downloaded directly via the Internet.
* Many SDKs are provided for free to encourage developers to use the system or language. Sometimes this is used as a marketing tool.

**ADD-ON**

* An SDK for an operating system add-on (for instance, QuickTime for Mac OS) include the add-on software itself, to be used for development purposes
* platforms where it is possible to develop applications that can at least start up on a system configuration without the add-on installed, and use a Gestalt-style run-time environment query to determine whether the add-on is present, and ones where the application will simply fail to start.

**PROVIDERS FOR SDK**

* Providers of SDKs for specific systems or subsystems may sometimes substitute a more specific term instead of software. For instance, both Microsoft and Apple provide driver development kits (DDK) for developing device material.



**ANDROID SOFTWARE DEVELOPMENT KIT**

* Cell phones are small in size and can easily be carried everywhere like a wallet
* Portability plays an important role in one’s life
* New and innovative functionalities are launching everyday
* ***Examples:*** *many handsets provide radio, TV reception, timer and clock, Internet Access and Camera and so on.*
* Developed by app developers and bundled with the OS.
* Interesting is third-party developed applications and android application development tools have been well received.
* The Android SDK is a mobile application development framework
* The SDK provides the tools and libraries necessary to develop app’s and that can run on Android-based devices
* The Android SDK is a mobile application development framework
* The SDK provides the tools and libraries necessary to develop app’s and that can run on Android-based devices

**ADVANTAGES:**

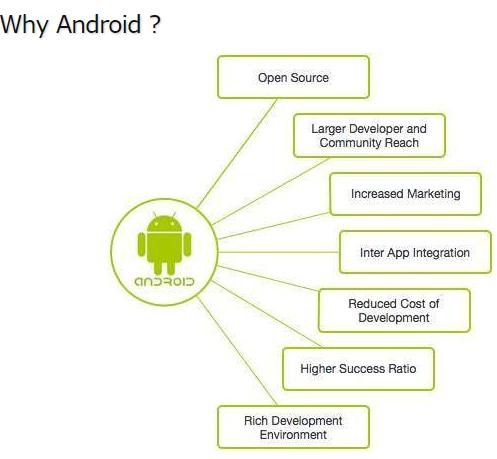
* 1. Android SDK is the low processor and RAM requirements
  2. Android SDK can be installed on all OS such as Windows, Mac OS and Linux
* Android SDK converts Java byte code to Android’s Dalvik VM byte code
* Finally Dalvik VM byte code runs the application developed by an developer
* Execution environments that are the derivation of JVM.
* To develop an Android app Environment needed are
  + 1. Android SDK (automatically install as a plug-in)
    2. The IDE Eclipse
    3. The Java Development Kit (JDK)

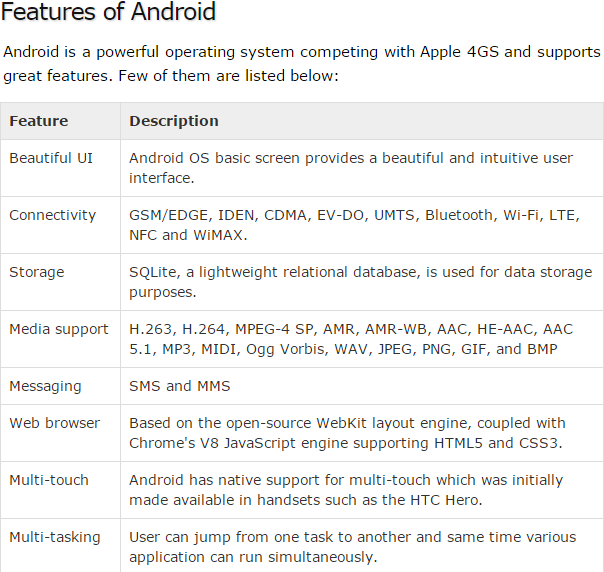
**FEATURES OF SDK**

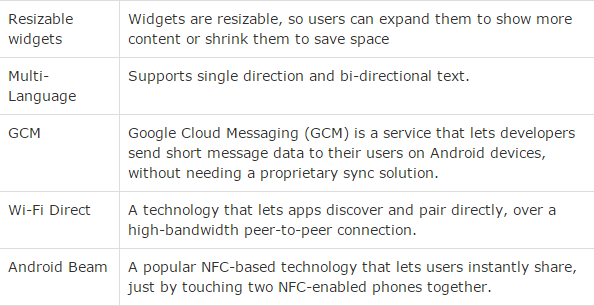
* Using the SDK we can either run the application on either
* Android device (or) A Software Emulator on the host machine

This both are achieved by using the Android Debug Bridge (ADB) available with the SDK. ADB is a client – server program includes 3 components

1. Host machine (developer machine)can invoke a client from a shell by giving an adb command
2. A daemon program which runs a background process on each emulator (which is a part manages communication with the handset and emulator)
3. A server program which runs the background process on the host machine(server manages communication b/w client & adb daemon)that runs on the emulator

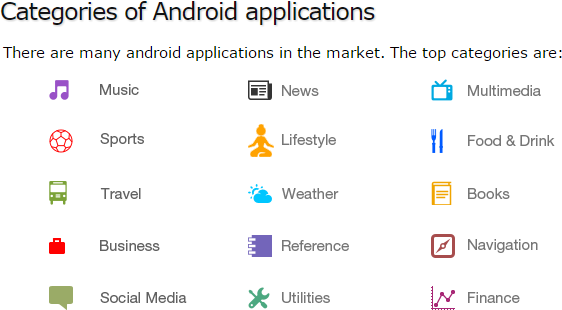






**ANDROID APPLICATIONS**

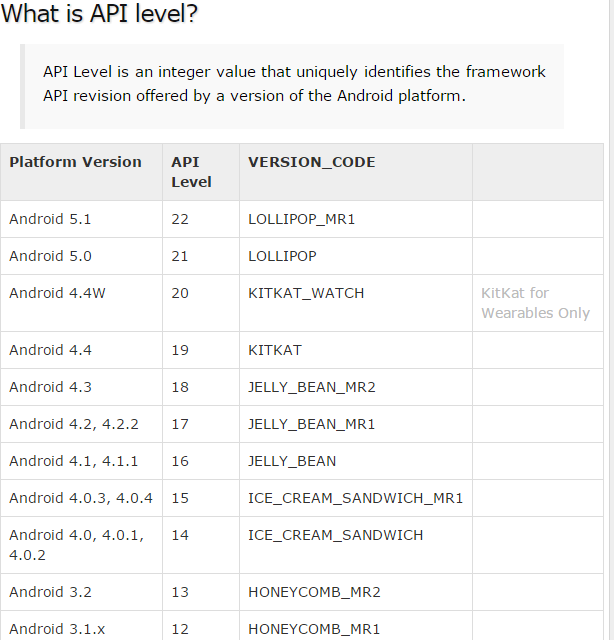
Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as **Google Play**,**SlideME**,**Opera Mobile Store**,**Mobango**,**F- droid** and the **Amazon Appstore**.

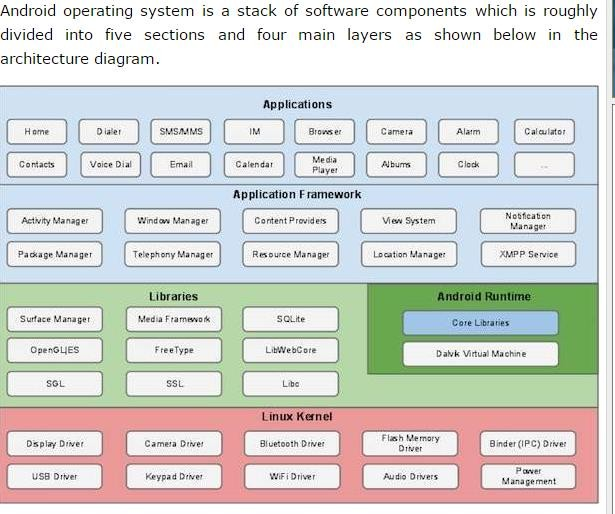


## History of Android:

The code names of Android ranges from A to L. Currently, such as Aestro, Blender, Cupcake, Donut, Éclair, Froyo, Gingerbread, Honeycomb, Ice cream Sandwitch, Jelly Bean, Kitkat and Lollipop. Lets understand the Android history in a sequence.





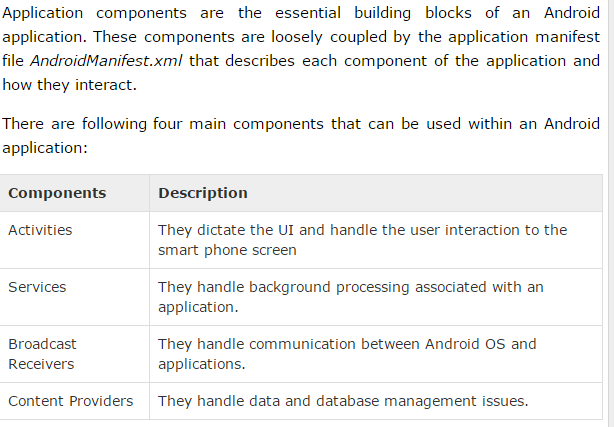


**APPLICATION FRAMEWORK**

The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

The Android framework includes the following key services −

* 1. **Activity Manager** − Controls all aspects of the application lifecycle and activity stack.
  2. **Content Providers** − Allows applications to publish and share data with other applications.
  3. **Resource Manager** − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
  4. **Notifications Manager** − Allows applications to display alerts and notifications to the user
  5. **View System** − An extensible set of views used to create application user interfaces.



## Activities

An activity represents a single screen with a user interface, in-short Activity performs actions on the screen.

For example, an **email application** might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails.

If an application has more than one activity, then one of them should be marked as the activity that is presented when the application is launched.

An activity is implemented as a subclass of Activity class as follows − public class MainActivity extends Activity {

}

## Services

A service is a component that runs in the background to perform long-running operations.

For example, a service might play music in the background while the user is in a different application, or it might fetch data over the network without blocking user interaction with an activity.

A service is implemented as a subclass of Service class as follows − public class MyService extends Service {

}

## Broadcast Receivers

Broadcast Receivers simply respond to broadcast messages from other applications or from the system.

For example, applications can also initiate broadcasts to let other applications know that some data has been downloaded to the device and is available for them to use, so this is broadcast receiver who will intercept this communication and will initiate appropriate action.

A broadcast receiver is implemented as a subclass of BroadcastReceiver class and each message is broadcaster as an Intent object.

public class MyReceiver extends BroadcastReceiver {

public void onReceive(context,intent){}

}

## Content Providers

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the **ContentResolve**r class.

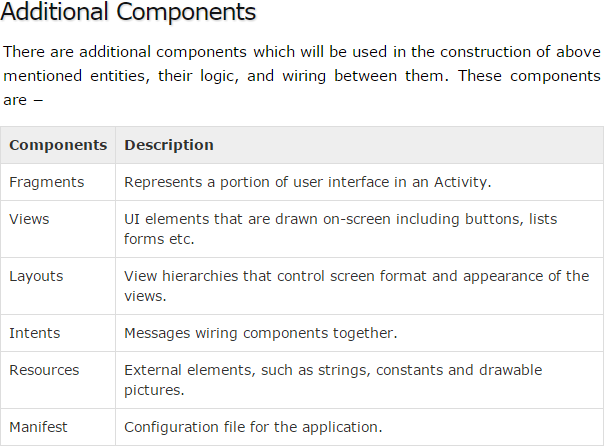
The data may be stored in the file system, the database or somewhere else entirely.

A content provider is implemented as a subclass of **ContentProvide**r class and must implement a standard set of APIs that enable other applications to perform transactions.

public class MyContentProvider extends ContentProvider {

public void onCreate(){}

}



# ANDROID CORE BUILDING BLOCKS

An android **component** is simply a piece of code that has a well defined life cycle e.g. Activity, Receiver, Service etc.

The **core building blocks** or **fundamental components** of android are

1. activities
2. views
3. intents
4. services
5. content providers
6. fragments
7. AndroidManifest.xml

## Activity

An activity is a class that represents a single screen. It is like a Frame in AWT.

## View

A view is the UI element such as button, label, text field etc. Anything that you see is a view.

## Intent

Intent is used to invoke components. It is mainly used to:

Start the service Launch an activity Display a web page

Display a list of contacts Broadcast a message Dial a phone call etc.

## Service

Service is a background process that can run for a long time. There are two types of services local and remote.

Local service is accessed from within the application whereas remote service is accessed remotely from other applications running on the same device.

## Content Provider

Content Providers are used to share data between the applications.

## Fragment

Fragments are like parts of activity. An activity can display one or more fragments on the screen at the same time.

## AndroidManifest.xml

It contains informations about activities, content providers, permissions etc. It is like the web.xml file in Java EE.

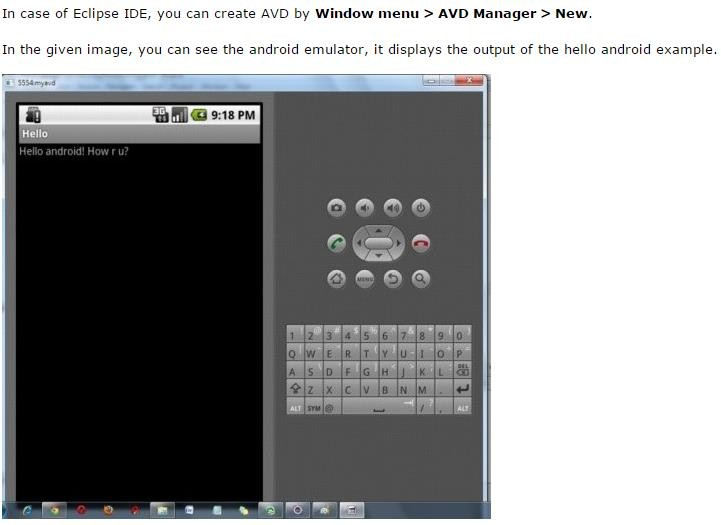
**ANDROID VIRTUAL DEVICE (AVD)**

It is used to test the android application without the need for mobile or tablet etc. It can be created in different configurations to emulate different types of real devices.

Android Emulator

**Android Emulator** is used to run, debug and test the android application.

If you don't have the real device, it can be the best way to run, debug and test the application. It uses an open source processor emulator technology called **QEMU**.



**ANDROID ACTIVITY LIFECYCLE**

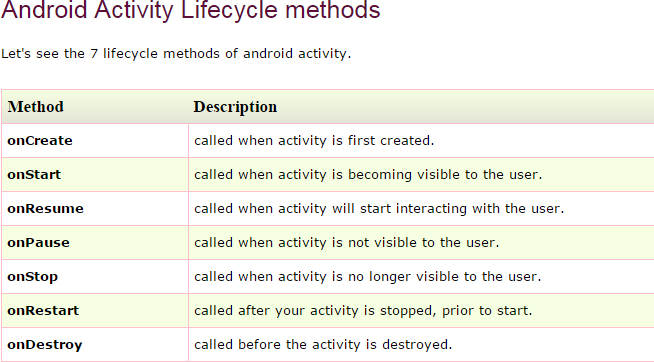
**Android Activity Lifecycle** is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class.

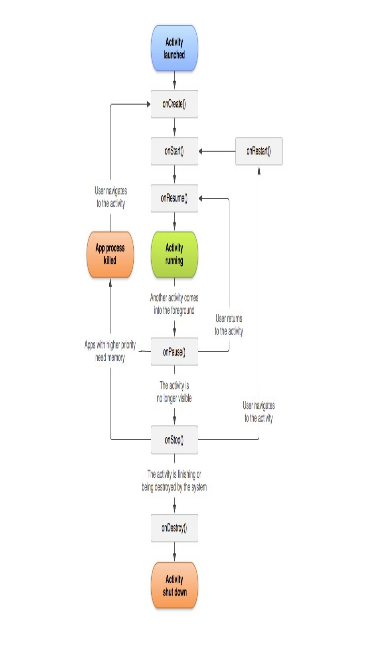
An activity is the single screen in android. It is like window or frame of Java.

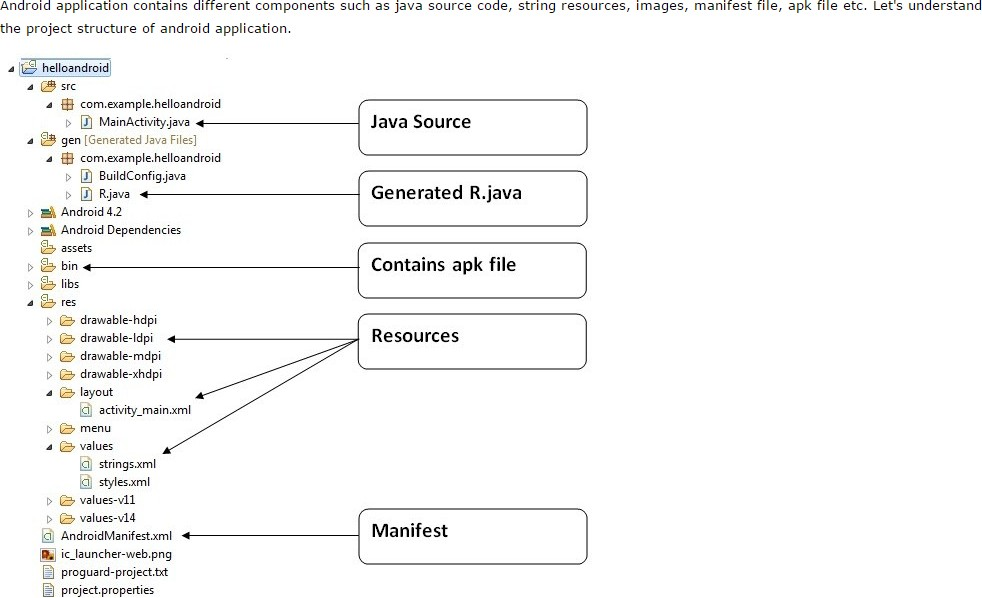
By the help of activity, you can place all your UI components or widgets in a single screen

.

The 7 lifecycle method of Activity describes how activity will behave at different states.









1. **Activity** is a java class that creates and default window on the screen where we can place different components such as Button, EditText, TextView, Spinner etc.

It is like the Frame of Java AWT.

It provides life cycle methods for activity such as onCreate, onStop, OnResume etc.

1. The **onCreate method** is called when Activity class is first created.
2. The **setContentView(R.layout.activity\_main)** gives information about our layout resource. Here, our layout resources are defined in activity\_main.xml file.

*File: activity\_main.xml*

**<RelativeLayout** xmlns:androclass=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent"

android:layout\_height="match\_parent" tools:context=".MainActivity" **>**

**<TextView** android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_centerHorizontal="true" android:layout\_centerVertical="true" android:text="@string/hello\_world" **/>**

## </RelativeLayout>

As you can see, a textview is created by the framework automatically. But the message for this string is defined in the strings.xml file.

The**@string/hello\_world** provides information about the textview message. The value of the attribute hello\_world is defined in the strings.xml file.

*File: strings.xml*

**<?xml** version="1.0" encoding="utf-8"**?>**

## <resources>

**<string** name="app\_name"**>**helloandroid**</string>**

**<string** name="hello\_world"**>**Hello world!**</string>**

**<string** name="menu\_settings"**>**Settings**</string>**

## </resources>

You can change the value of the hello\_world attribute from this file.

**Generated R.java file**

It is the auto-generated file that contains IDs for all the resources of res directory. It is generated by aapt(Android Asset Packaging Tool).

Whenever you create any component on activity\_main, a corresponding ID is created in the R.java file which can be used in the Java Source file later.

**APK File**

An apk file is created by the framework automatically. If you want to run the android application on the mobile, transfer and install it.

**Resources**

It contains resource files including activity\_main, strings, styles etc.

**Manifest file**

It contains information about package including components such as activities, services, content providers etc.

Whatever component you develop as a part of your application, you must declare all its components in a *manifest.xml* which resides at the root of the application project directory.

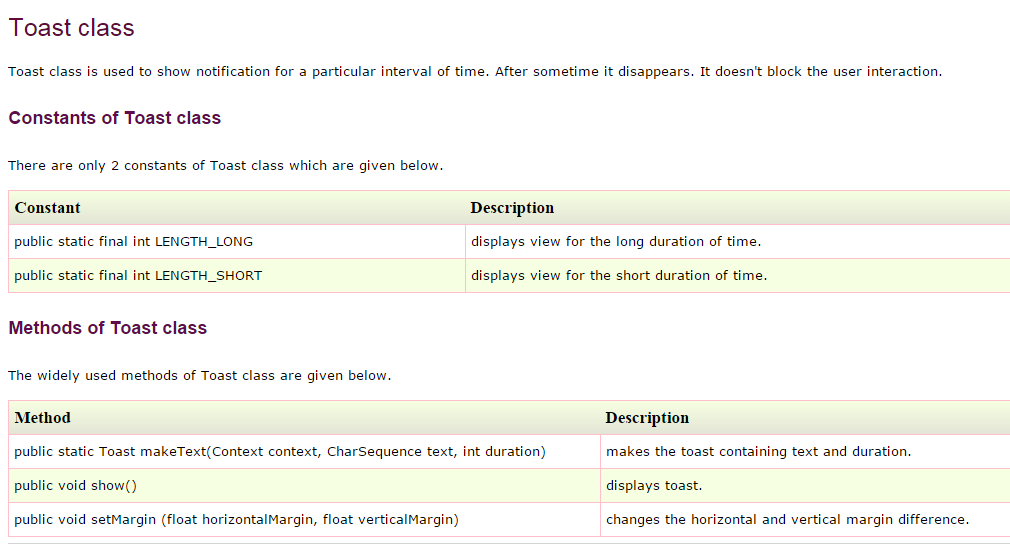
This file works as an interface between Android OS and your application, so if you do not declare your component in this file, then it will not be considered by the OS

Andorid Toast can be used to display information for the short period of time.

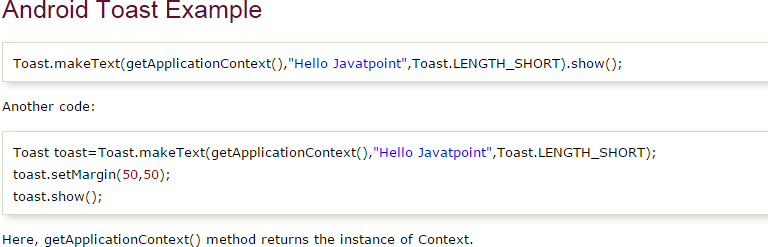
A toast contains message to be displayed quickly and disappears after sometime.

**Toast class**

Toast class is used to show notification for a particular interval of time. After sometime it disappears. It doesn't block the user interaction.



22



**Android Intent** is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.

It is generally used with startActivity() method to invoke activity, broadcast receivers etc. Android intents are mainly used to:

Start the service Launch an activity Display a web page

Display a list of contacts Broadcast a message Dial a phone call etc.

Types of Android Intents

There are two types of intents in android: implicit and explicit.

* 1. Implicit Intent

**Implicit Intent** doesn't specify the component.

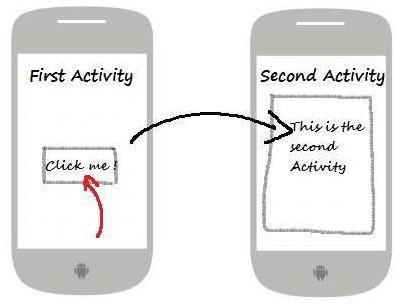
In such case, intent provides information of available components provided by the system that is to be invoked.

For example, you may write the following code to view the webpage. **Intent intent=new Intent(Intent.ACTION\_VIEW); intent.setData(Uri.parse**[**("h**](http://www.javatpoint.com/)**tt**[**p://www.javatpoint.com**](http://www.javatpoint.com/)**")); startActivity(intent);**

* 1. Explicit Intent

**Explicit Intent** specifies the component. In such case, intent provides the external class to be invoked.

## Intent i = new Intent(getApplicationContext(), ActivityTwo.class); startActivity(i);



**EXNO: 2 HELLO WORLD DATE:**

**AIM**

To develop an android application to display “hello world”.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:tools=["http://schemas.android.com/tool](http://schemas.android.com/tools)s" android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingBottom=*"@dimen/activity\_vertical\_margin"* android:paddingLeft=*"@dimen/activity\_horizontal\_margin"* android:paddingRight=*"@dimen/activity\_horizontal\_margin"* android:paddingTop=*"@dimen/activity\_vertical\_margin"* tools:context=*".MainActivity"*>

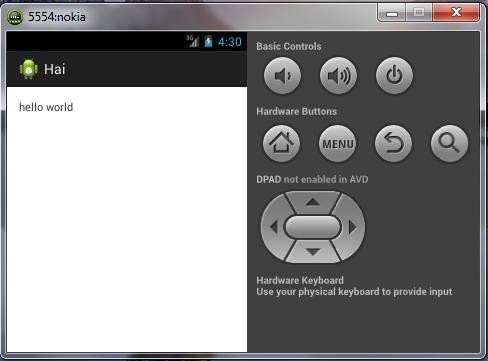
<TextView

android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"*

## android:text=*"*hello world*"* />

</RelativeLayout>

**OUTPUT:**



**RESULT:**

Thus the Application was developed successfully.

## EX NO: 3 COMPONENT INTENT DATE:

**AIM**

To develope the android application using component intent.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

package com.example.intent; import android.net.Uri; import android.os.Bundle; import android.app.Activity; import android.content.Intent; import android.view.Menu;

public class MainActivity extends Activity { @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

**Intent x=new Intent(Intent.ACTION\_VIEW,Uri.parse("http**[**s://www.google.c**](http://www.google.co.in/)**o.in")); startActivity(x);**

}@Override

public boolean onCreateOptionsMenu(Menu menu) {

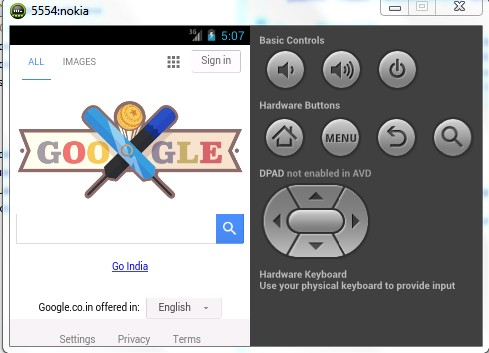
// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.main, menu);

return true;

}

}

**OUTPUT:**



**RESULT:**

Thus the Application was developed successfully**.**

## EX NO: 4 TOAST DATE:

**AIM:**

To develope the android application using toast function.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

package com.example.toast;

import android.os.Bundle; import android.app.Activity; import android.view.Menu; import android.widget.Toast;

publicclass MainActivity extends Activity { @Override

protectedvoid onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*); **Toast.makeText(getApplicationContext(), "MESSAGE SENT", Toast.LENGTH\_LONG).show();**

}

@Override

publicboolean onCreateOptionsMenu(Menu menu) {

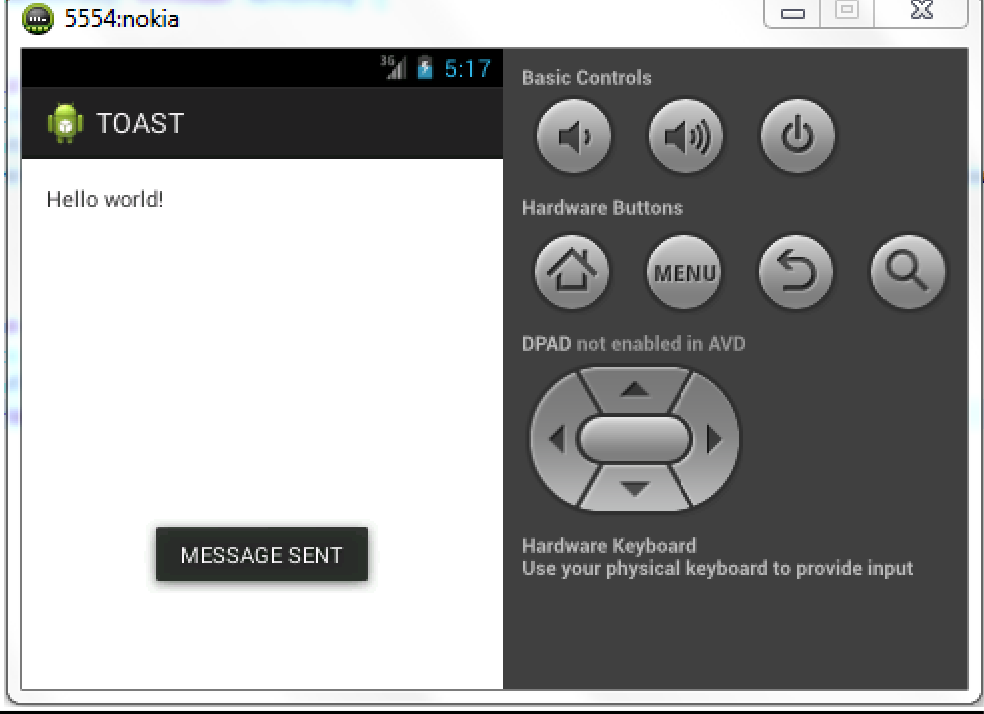
// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.main, menu);

returntrue;

}

}

**OUTPUT:**



**RESULT:**

Thus the Application was developed successfully.

## EX NO: 5 GUI COMPONENTS / LOGIN APPLICATION DATE:

**AIM:**

To develope the android application that uses GUI Components.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Activity\_main.xml:**

## Buttons: b Edittext:e1,e2.

<RelativeLayout xmlns:android=[*"http://schemas.android.com/apk/res/androi*](http://schemas.android.com/apk/res/android)*d"* xmlns:tools=[*"http://schemas.android.com/tools"*](http://schemas.android.com/tools)android:layout\_width=*"match\_parent"* android:layout\_height=*"match\_parent"* android:paddingBottom=*"@dimen/activity\_vertical\_margin"* android:paddingLeft=*"@dimen/activity\_horizontal\_margin"* android:paddingRight=*"@dimen/activity\_horizontal\_margin"* android:paddingTop=*"@dimen/activity\_vertical\_margin"* tools:context=*".MainActivity"* >

**<TextView android:id=*"@+id/textView2"* android:layout\_width=*"wrap\_content"***

**android:layout\_height=*"wrap\_content"* android:text=*"Login page"* />**

**<TextView android:id=*"@+id/textView1"* android:layout\_width=*"wrap\_content"***

**android:layout\_height=*"wrap\_content"* android:layout\_alignLeft=*"@+id/textView2"* android:layout\_below=*"@+id/textView2"* android:layout\_marginLeft=*"34dp"* android:layout\_marginTop=*"44dp"* android:text=*"user name"* />**

**<TextView android:id=*"@+id/textView3"* android:layout\_width=*"wrap\_content"***

**android:layout\_height=*"wrap\_content"* android:layout\_alignRight=*"@+id/textView1"* android:layout\_below=*"@+id/textView1"* android:layout\_marginTop=*"28dp"* android:text=*"password"* />**

**<Button android:id=*"@+id/b"***

**android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"* android:layout\_centerHorizontal=*"true"* android:layout\_centerVertical=*"true"* android:text=*"login"* />**

**<EditText android:id=*"@+id/e1"***

**android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"* android:layout\_alignLeft=*"@+id/button1"* android:layout\_alignTop=*"@+id/textView1"* android:layout\_marginLeft=*"27dp"* android:ems=*"10"* >**

## <requestFocus />

**</EditText>**

**<EditText android:id=*"@+id/e2"***

**android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"* android:layout\_alignLeft=*"@+id/editText1"* android:layout\_alignRight=*"@+id/editText1"* android:layout\_alignTop=*"@+id/textView3"* android:ems=*"10"* android:inputType=*"textPassword"* />**

</RelativeLayout>

**Mainactivity.java:**

**package** com.example.login; **import** android.os.Bundle; **import** android.app.Activity; **import** android.view.Menu; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.Toast;

**public class** MainActivity **extends** Activity { EditText ed1,ed2;

Button b; @Override

**protected void** onCreate(Bundle savedInstanceState) { **super**.onCreate(savedInstanceState); setContentView(R.layout.*activity\_main*); **ed1=(EditText)findViewById(R.id.*e1*); ed2=(EditText)findViewById(R.id.*e2*); b=(Button)findViewById(R.id.*b*); b.setOnClickListener(new View.OnClickListener() {**

## public void onClick(View v) { String a=ed1.getText().toString(); String b=ed2.getText().toString();

**if(a.equals("welcome")&&b.equals("abcd"))**

## {

**Toast.*makeText*(getApplicationContext(), "success", Toast.*LENGTH\_LONG*).show();**

## }

**else**

## {

**Toast.*makeText*(getApplicationContext(), "Invalid user", Toast.*LENGTH\_LONG*).show();**

## }

**}});**

}

**protected void** onReserve()

{

**super**.onResume();

}

@Override

**public boolean** onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.*main*, menu);

## return true;

}

}

**OUTPUT:**



**RESULT:**

Thus the Application was developed successfully.

## EX NO: 6 CALCULATOR DATE:

**AIM:**

To develop an android application using calculator function.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Activity\_main.xml:**

## Button:+,-,\*,/. Edittext:etNum1,etNum2.

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:andr[oid="htt](http://schemas.android.com/apk/res/android)p:/[/sc](http://schemas.android.com/apk/res/android)he[mas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android) android:orientation="vertical" android:layout\_width="fill\_parent" android:layout\_height="fill\_parent">

<LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/linearLayout1" android:layout\_marginLeft="10pt" android:layout\_marginRight="10pt" android:layout\_marginTop="3pt">

## <EditText android:layout\_weight="1"

**android:layout\_height="wrap\_content" android:layout\_marginRight="5pt" android:id="@+id/etNum1" android:layout\_width="match\_parent" android:inputType="numberDecimal">**

## </EditText>

**<EditText android:layout\_height="wrap\_content" android:layout\_weight="1" android:layout\_marginLeft="5pt" android:id="@+id/etNum2" android:layout\_width="match\_parent" android:inputType="numberDecimal">**

## </EditText>

**</LinearLayout>**

## <LinearLayout android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:id="@+id/linearLayout2" android:layout\_marginTop="3pt" android:layout\_marginLeft="5pt" android:layout\_marginRight="5pt">

**<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="+" android:textSize="8pt" android:id="@+id/btnAdd">**

## </Button>

**<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="-" android:textSize="8pt" android:id="@+id/btnSub">**

## </Button>

**<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="\*" android:textSize="8pt" android:id="@+id/btnMult">**

## </Button>

**<Button android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_weight="1" android:text="/" android:textSize="8pt" android:id="@+id/btnDiv">**

## </Button>

</LinearLayout>

<TextView android:layout\_height="wrap\_content" android:layout\_width="match\_parent" android:layout\_marginLeft="5pt" android:layout\_marginRight="5pt" android:textSize="12pt" android:layout\_marginTop="3pt" android:id="@+id/tvResult" android:gravity="center\_horizontal">

</TextView>

</LinearLayout>

**Mainactivity.java:**

package com.website.androidcalculator; import android.os.Bundle;

import android.app.Activity; import android.text.TextUtils; import android.view.Menu; import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText; import android.widget.TextView;

public class MainActivity extends Activity implements OnClickListener { EditText etNum1;

EditText etNum2;

Button btnAdd; Button btnSub; Button btnMult;

Button btnDiv;

TextView tvResult;

String oper = "";

/\*\* Called when the activity is first created. \*/ @Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

// find the elements

## etNum1 = (EditText) findViewById(R.id.etNum1); etNum2 = (EditText) findViewById(R.id.etNum2);

**btnAdd = (Button) findViewById(R.id.btnAdd); btnSub = (Button) findViewById(R.id.btnSub); btnMult = (Button) findViewById(R.id.btnMult); btnDiv = (Button) findViewById(R.id.btnDiv);**

## tvResult = (TextView) findViewById(R.id.tvResult);

// set a listener **btnAdd.setOnClickListener((OnClickListener) this); btnSub.setOnClickListener(this); btnMult.setOnClickListener(this); btnDiv.setOnClickListener(this);**

}

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

## float num1 = 0; float num2 = 0; float result = 0;

**// check if the fields are empty**

## if (TextUtils.isEmpty(etNum1.getText().toString())

**|| TextUtils.isEmpty(etNum2.getText().toString())) { return;**

## }

**// read EditText and fill variables with numbers**

## num1 = Float.parseFloat(etNum1.getText().toString());

**num2 = Float.parseFloat(etNum2.getText().toString());**

## // defines the button that has been clicked and performs the corresponding operation

**// write operation into oper, we will use it later for output switch (v.getId()) {**

## case R.id.btnAdd: oper = "+";

**result = num1 + num2; break;**

## case R.id.btnSub: oper = "-";

**result = num1 - num2; break;**

## case R.id.btnMult: oper = "\*";

**result = num1 \* num2; break;**

## case R.id.btnDiv: oper = "/";

**result = num1 / num2; break;**

## default: break;

**}**

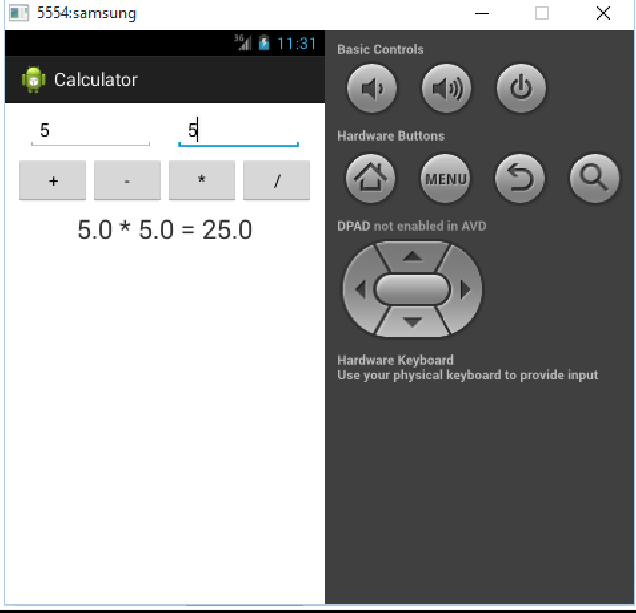
## // form the output line

**tvResult.setText(num1 + " " + oper + " " + num2 + " = " + result);**

## }

}

**OUTPUT:**



**RESULT:**

Thus the Application was developed successfully.

## EX NO: 7 RADIOBUTTON DATE:

**AIM:**

To develop an android application that uses radio button to change the font color , size and background color.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Activity\_main.xml: Radiobutton:radiobutton1,radiobutton2,radiobutton3.**

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent"

android:id="@+id/gsat" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView

android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:text="@string/hello\_world" />

<RadioButton

android:id="@+id/radioButton1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/textView1" android:layout\_below="@+id/textView1" android:layout\_marginLeft="24dp" android:layout\_marginTop="28dp" android:onClick="click" android:text="Change Text Color" />

<RadioButton

android:id="@+id/radioButton2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/radioButton1" android:layout\_below="@+id/radioButton1" android:layout\_marginTop="18dp" android:onClick="click"

android:text="Change Background" />

<RadioButton

android:id="@+id/radioButton3" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/radioButton2" android:layout\_below="@+id/radioButton2" android:layout\_marginLeft="16dp" android:layout\_marginTop="18dp" android:onClick="click" android:text="Increase the Font Size" />

</RelativeLayout>

**Mainactivity.java:**

package radio.button;

import radio.button.R.dimen; import android.R.color; import android.os.Bundle; import android.app.Activity;

import android.graphics.Typeface;

import android.support.v4.widget.SimpleCursorAdapter.ViewBinder;

import android.util.TypedValue; import android.view.Menu; import android.view.View;

import android.view.View.OnClickListener; import android.widget.RadioButton;

import android.widget.RelativeLayout; import android.widget.TextView; import android.widget.Toast;

public class MainActivity extends Activity implements View.OnClickListener{ private RadioButton rb1,rb2,rb3;

private TextView tv1; private RelativeLayout rl;

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main); **rb1=(RadioButton)findViewById(R.id.radioButton1); rb2=(RadioButton)findViewById(R.id.radioButton2); rb3=(RadioButton)findViewById(R.id.radioButton3); tv1=(TextView)findViewById(R.id.textView1); rl=(RelativeLayout)findViewById(R.id.gsat);**

}

public void click(View v){

## boolean on=((RadioButton)v).isChecked(); switch(v.getId()){

**case R.id.radioButton1: if(on){**

## //tv1.setBackgroundColor(0xff00ff00); tv1.setTextColor(0xfffff0ff);

**}**

## break;

**case R.id.radioButton2: if(on){**

## rl.setBackgroundResource(R.drawable.ic\_launcher);

**}**

## break;

**case R.id.radioButton3: tv1.setClickable(true);**

## int ht=(int)TypedValue.applyDimension(TypedValue.COMPLEX\_UNIT\_DIP, 5,getResources().getDisplayMetrics());

**Typeface face=Typeface.DEFAULT\_BOLD; int typ=TypedValue.COMPLEX\_UNIT\_IN; tv1.setTextScaleX(typ);**

## tv1.setTextSize(ht); tv1.setTypeface(face);

**Toast.makeText(getApplicationContext(), "Sathish",Toast.LENGTH\_LONG).show();**

## break;

**}**

## rb1.setChecked(false); rb2.setChecked(false); rb3.setChecked(false);

**}**

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@Override

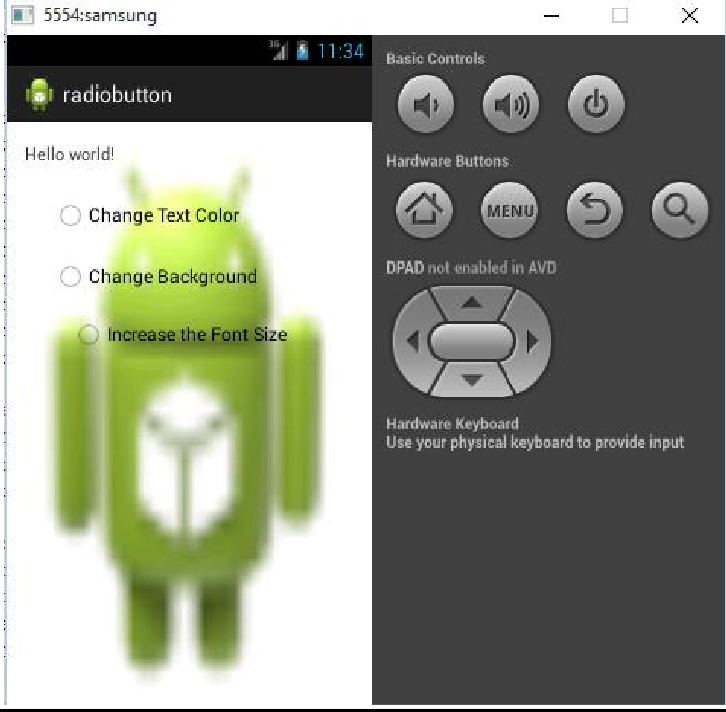
public void onClick(View v) {

// TODO Auto-generated method stub

}

}

**OUTPUT:**



**RESULT:**

Thus the Application was developed successfully.

## EX NO: 8 GRAPHICAL PRIMITIVES DATE:

**AIM:**

To develop an android application using graphical primitives.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Mainactivity.java:**

package com.example.simplegraphics; import android.os.Bundle;

import android.app.Activity;

import android.view.Menu; import android.content.Context; import android.graphics.Canvas; import android.graphics.Color; import android.graphics.Paint; import android.view.View;

public class MainActivity extends Activity { DemoView demoview;

/\*\* Called when the activity is first created. \*/ @Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState); demoview = new DemoView(this); setContentView(demoview);

}

private class DemoView extends View{ public DemoView(Context context){

super(context);

}

@Override protected void onDraw(Canvas canvas) { super.onDraw(canvas);

// custom drawing code here Paint paint = new Paint(); paint.setStyle(Paint.Style.FILL);

// make the entire canvas white paint.setColor(Color.WHITE); canvas.drawPaint(paint);

// draw blue circle with anti aliasing turned off paint.setAntiAlias(false); paint.setColor(Color.BLUE); canvas.drawCircle(20, 20, 15, paint);

// draw green circle with anti aliasing turned on paint.setAntiAlias(true); paint.setColor(Color.GREEN); canvas.drawCircle(60, 20, 15, paint);

// draw red rectangle with anti aliasing turned off paint.setAntiAlias(false); paint.setColor(Color.RED); canvas.drawRect(100, 5, 200, 30, paint);

// draw the rotated text canvas.rotate(-45);

paint.setStyle(Paint.Style.FILL); canvas.drawText("Graphics Rotation", 40, 180, paint);

//undo the rotate canvas.restore();

}

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

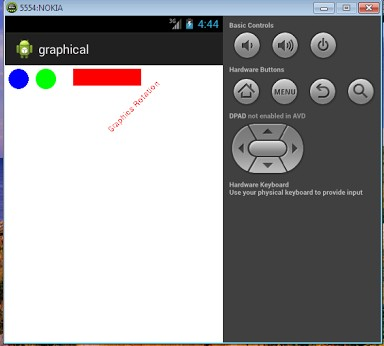
// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.main, menu);

return true;

}

}

**OUTPUT:**



**RESULT:**

Thus the Application was developed successfully.

## EX NO: 9 DATABASE DATE:

**AIM:**

To store the student details in a database and view the details using eclipse.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Activity\_main.xml:**

## Edittext:edittext1.

**Button:button1,button2.**

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView

**android:id="@+id/textView1"** android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="@string/hello\_world" />

## <EditText

**android:id="@+id/editText1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/textView1" android:layout\_below="@+id/textView1" android:layout\_marginLeft="17dp" android:ems="10" >**

## <requestFocus />

**</EditText>**

## <Button

**android:id="@+id/button1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignLeft="@+id/editText1" android:layout\_below="@+id/editText1" android:layout\_marginTop="38dp" android:text="add" />**

## <Button

**android:id="@+id/button2" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignBaseline="@+id/button1" android:layout\_alignBottom="@+id/button1" android:layout\_alignParentRight="true" android:layout\_marginRight="36dp" android:text="view" />**

</RelativeLayout>

**Main\_activity.java:**

package test.sql;

import android.annotation.SuppressLint; import android.app.Activity;

import android.app.AlertDialog; import android.content.Context; import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase; import android.os.Bundle;

import android.view.Menu; import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText; import android.widget.Toast;

## public class MainActivity extends Activity implements OnClickListener

**{**

## EditText et1;

**Button b1,b2; SQLiteDatabase DB;**

## @SuppressLint("InlinedApi")

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main); **et1=(EditText)findViewById(R.id.editText1); b1=(Button)findViewById(R.id.button1); b2=(Button)findViewById(R.id.button2);**

## DB=openOrCreateDatabase("GSAT",Context.MODE\_PRIVATE,null); DB.execSQL("CREATE TABLE IF NOT EXISTS student(name VARCHAR);");

**b1.setOnClickListener(this); b2.setOnClickListener(this);**

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.main, menu);

return true;

}

@Override

## public void onClick(View arg0) {

**// TODO Auto-generated method stub if(arg0==b1){**

## //Intent ab=new Intent(this,list.class);

**//startActivity(ab);**

## Toast.makeText(getApplicationContext(), "BUTTON ADD", Toast.LENGTH\_LONG).show(); if(et1.getText().toString().trim().length()==0){ showmessage("error","Enter valid Input");

**}**

## DB.execSQL("INSERT INTO Student VALUES('"+et1.getText()+"');"); showmessage("Sucess","Name Added");

**clear();**

## }

**if(arg0==b2){**

## Cursor c=DB.rawQuery("SELECT \* FROM student; ",null ); if(c.getCount()==0){

**showmessage("...","No record Found"); return;**

## }

**StringBuffer buffer=new StringBuffer(); while(c.moveToNext())**

## {

**buffer.append("Name: "+c.getString(0)+"\n");**

## }

**showmessage("Student Details", buffer.toString());**

## }

**}**

## private void clear() {

**// TODO Auto-generated method stub et1.clearFocus();**

## }

**private void showmessage(String string, String string2) {**

## // TODO Auto-generated method stub

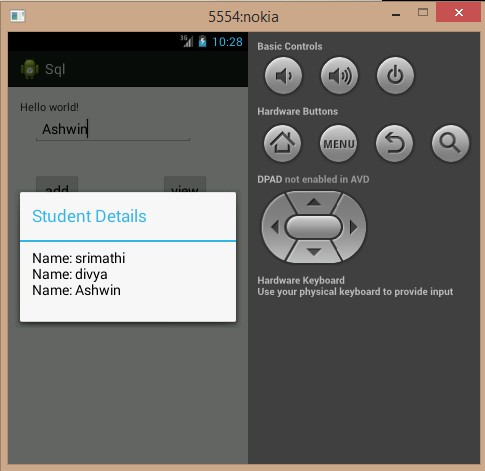
**AlertDialog.Builder builder=new AlertDialog.Builder(this); builder.setCancelable(true);**

## builder.setTitle(string); builder.setMessage(string2); builder.show();

**}**

## }

**OUTPUT:**



**RESULT:**

Thus the program is executed successfully.

## EX NO: 10 GLOBAL POSITIONING SYSTEM (GPS) DATE:

**AIM:**

To find the location of a person using GPS.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Activity\_main.xml:**

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView

**android:id="@+id/textView1"** android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" **android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true" android:layout\_marginTop="166dp"**

## android:text="Large Text" android:textAppearance="?android:attr/textAppearanceLarge" />

<TextView

**android:id="@+id/textView2"** android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

## android:layout\_alignLeft="@+id/textView1" android:layout\_below="@+id/textView1" android:layout\_marginTop="69dp" android:text="Medium Text"

**android:textAppearance="?android:attr/textAppearanceMedium" />**

</RelativeLayout>

**Main Activity.java:**

package com.example.hari; import android.os.Bundle; import android.app.Activity; import android.view.Menu; **import java.util.List; import android.R.string;**

## import android.location.Location;

**import android.location.LocationListener; import android.location.LocationManager; import android.content.Context;**

## import android.widget.TextView;

**public class MainActivity extends Activity implements LocationListener { LocationManager m;**

## String str="iojk"; TextView t1,t2;

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_main);

## m=(LocationManager) getSystemService(Context.LOCATION\_SERVICE); m.requestLocationUpdates(LocationManager.GPS\_PROVIDER, 1, 1000, this); t1=(TextView)findViewById(R.id.textView1); t2=(TextView)findViewById(R.id.textView2);

}

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.main, menu);

return true;

}

## public void onLocationChanged(Location arg0)

**{**

## t1.setText("latitude"+ arg0.getLatitude()); t2.setText("longitude:"+arg0.getLongitude());

**// TODO Auto-generated method stub**

## }

**public void onProviderDisabled(String arg0)**

## {

**// TODO Auto-generated method stub**

## }

**public void onProviderEnabled(String arg0)**

## {

**// TODO Auto-generated method stub**

## }

**public void onStatusChanged(String arg0, int arg1, Bundle arg2)**

## {

**// TODO Auto-generated method stub**

## }

**}**

**AndroidManifest.xml:**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) package="com.example.hari"

android:versionCode="1" android:versionName="1.0" >

## <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/>

<uses-sdk

android:minSdkVersion="8" android:targetSdkVersion="18" />

<application

android:allowBackup="true" android:icon="@drawable/ic\_launcher" android:label="@string/app\_name" android:theme="@style/AppTheme" >

<activity

android:name="com.example.hari.MainActivity" android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

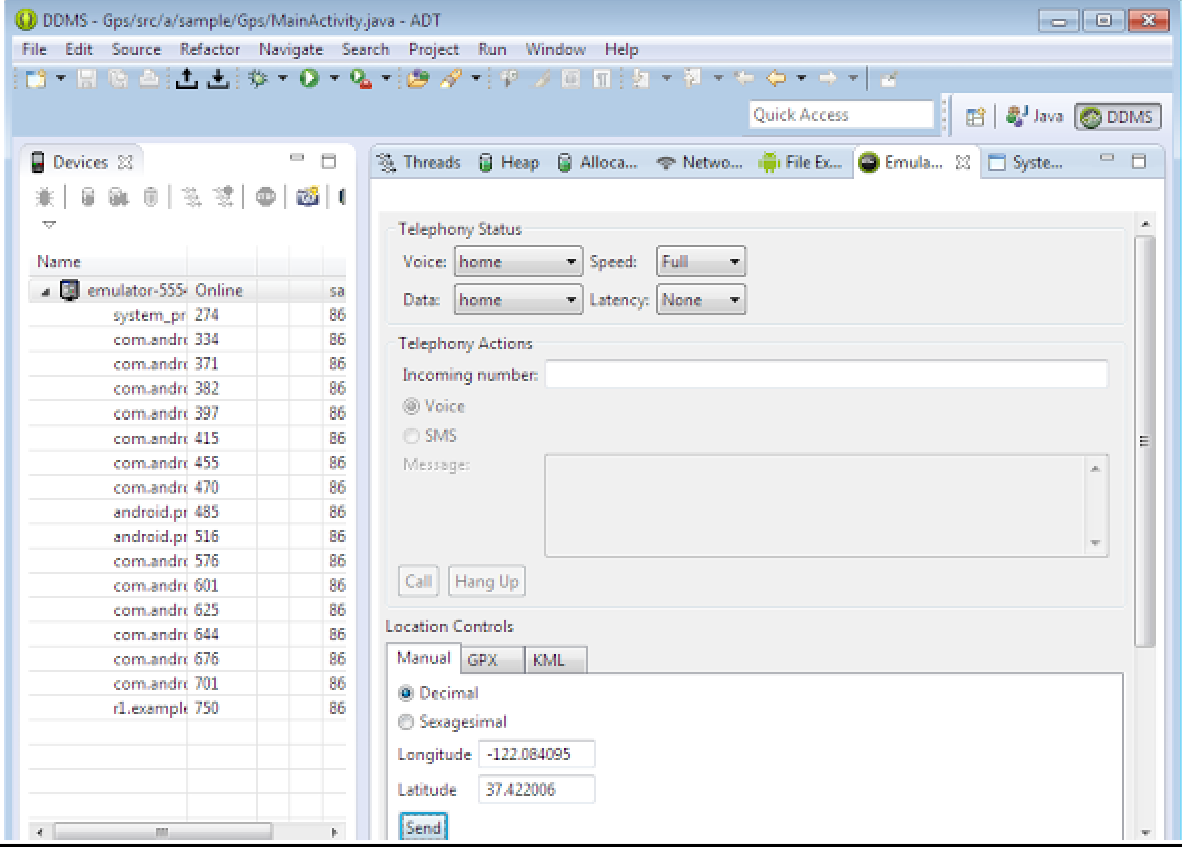
</intent-filter>

</activity>

</application>

</manifest>

**OUTPUT:**



**RESULT:**

Thus the program is executed successfully.

## EX NO: 11 SHORT MESSAGE SERVICE (SMS) DATE:

**AIM:**

To develop android application to send SMS.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Activity\_main.xml: EditText:editTextPhoneNo,editTextSMS.**

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".MainActivity" >

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"

android:text="@string/hello\_world" />

<Button

android:id="@+id/buttonSend" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentBottom="true" android:layout\_centerHorizontal="true" android:layout\_marginBottom="133dp" android:text="send" />

<EditText android:id="@+id/editTextPhoneNo" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_above="@+id/buttonSend"

android:layout\_alignLeft="@+id/editTextSMS" android:layout\_marginBottom="90dp" android:ems="10" >

<requestFocus />

</EditText>

<EditText android:id="@+id/editTextSMS" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_centerHorizontal="true" android:layout\_centerVertical="true" android:ems="10" />

</RelativeLayout>

**Main Activity.java:**

**package** m.shor;

**import** android.os.Bundle;

**import** android.app.Activity;

**import** android.telephony.SmsManager;

**import** android.view.Menu; **import** android.view.View; **import** android.widget.Button; **import** android.widget.EditText; **import** android.widget.Toast;

**public class** MainActivity **extends** Activity { Button buttonsend;

EditText et1,et2;

@Override

**protected void** onCreate(Bundle savedInstanceState) { **super**.onCreate(savedInstanceState); setContentView(R.layout.*activity\_main*);

**buttonsend = (Button)findViewById(R.id.*buttonSend*); et1 = (EditText)findViewById(R.id.*editTextPhoneNo*); et2 = (EditText)findViewById(R.id.*editTextSMS*);**

## buttonsend.setOnClickListener(new View.OnClickListener() {

@Override

**public void** onClick(View arg0) {

// **TODO** Auto-generated method stub **String phno = et1.getText().toString(); String sms = et2.getText().toString(); try**

## {

**SmsManager smsManager = SmsManager.*getDefault*();**

## smsManager.sendTextMessage(phno,null,sms,null,null);

**Toast.*makeText*(getApplicationContext(), "SMS Sent!",Toast.*LENGTH\_LONG*).show();**

## }

**catch(Exception e)**

## {

}

@Override

## }

}

});

**Toast.*makeText*(getApplicationContext(),**

## "SMS faild, please try again later!", Toast.*LENGTH\_LONG*).show();

**e.printStackTrace();**

**public boolean** onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.*main*, menu);

**return true**; }}

**SMS MANIFEST:**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<manifest xmlns:android=[*"http://schemas.android.com/apk/res/android"*](http://schemas.android.com/apk/res/android)package=*"m.shor"*

android:versionCode=*"1"* android:versionName=*"1.0"* >

<uses-sdk

android:minSdkVersion=*"8"* android:targetSdkVersion=*"17"* />

<uses-permission android:name=*"android.permission.SEND\_SMS"*/>

<application android:allowBackup=*"true"*

android:icon=*"@drawable/ic\_launcher"* android:label=*"@string/app\_name"* android:theme=*"@style/AppTheme"* >

<activity android:name=*"m.shor.MainActivity"* android:label=*"@string/app\_name"* >

<intent-filter>

<action android:name=*"android.intent.action.MAIN"* />

<category android:name=*"android.intent.category.LAUNCHER"* />

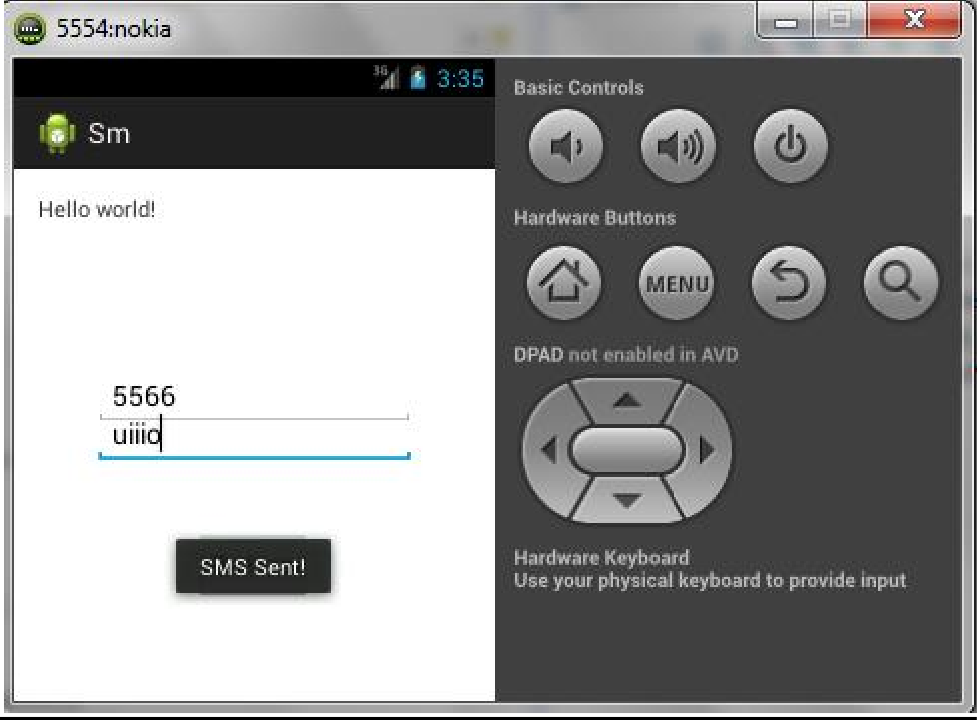
</intent-filter>

</activity>

</application>

</manifest>

**OUTPUT:**



**RESULT:**

Thus the program is executed successfully.

## EX NO: 12 ALARM DATE:

**AIM:**

To develop android application to Alarm.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name and select next.

1. Choose the android version.Choose the lowest android version(Android 2.2) and select next.
2. Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to res folder and select layout.Double click the main.xml file. 7)Now you can see the Graphics layout window.

8)Click the main.xml file and type the xml code in the respective window. 9)Click the mainactitvity.java and type the code.

**PROGRAM:**

**Activity\_main.xml:**

<RelativeLayout xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) xmlns:t[ools="ht](http://schemas.android.com/tools)tp:/[/sc](http://schemas.android.com/tools)he[mas.android.com/tools"](http://schemas.android.com/tools) android:layout\_width="match\_parent" android:layout\_height="match\_parent" android:paddingBottom="@dimen/activity\_vertical\_margin" android:paddingLeft="@dimen/activity\_horizontal\_margin" android:paddingRight="@dimen/activity\_horizontal\_margin" android:paddingTop="@dimen/activity\_vertical\_margin" tools:context=".Alaram" >

<TextView android:id="@+id/textView1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_alignParentTop="true" android:layout\_centerHorizontal="true"

android:layout\_marginTop="34dp" android:text="Enter Seconds" />

<EditText android:id="@+id/editText1"

android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/textView1" android:layout\_centerHorizontal="true" android:layout\_marginTop="48dp" android:ems="10" />

<Button

android:id="@+id/button1" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:layout\_below="@+id/editText1" android:layout\_centerHorizontal="true" android:layout\_marginTop="20dp" android:text="Button" />

</RelativeLayout>

## Main Activity.java:

package test.alarm; import android.os.Bundle;

import android.app.Activity;

import android.app.AlarmManager; import android.app.PendingIntent; import android.content.Intent; import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener; import android.widget.Button;

import android.widget.EditText; import android.widget.Toast;

public class Alaram extends Activity implements OnClickListener { Button b1;

EditText text; @Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState);

setContentView(R.layout.alaram); b1=(Button) findViewById(R.id.button1);

text = (EditText) findViewById(R.id.editText1); b1.setOnClickListener(this);

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present. getMenuInflater().inflate(R.menu.alaram, menu);

return true;

}

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

int i = Integer.parseInt(text.getText().toString()); Intent intent = new Intent(this, MyBroadcastReceiver.class); PendingIntent pendingIntent =

PendingIntent.getBroadcast(this.getApplicationContext(),100, intent, 0);

AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE); alarmManager.set(AlarmManager.RTC\_WAKEUP, System.currentTimeMillis() + (i \*

1000), pendingIntent);

Toast.makeText(this, "Alarm set in " + i + " seconds",Toast.LENGTH\_LONG).show();

}

}

**ALARM MANIFEST:**

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android=["http://schemas.android.com/](http://schemas.android.com/apk/res/android)a[pk/res/android"](http://schemas.android.com/apk/res/android) package="test.alarm"

android:versionCode="1" android:versionName="1.0" >

<uses-sdk android:minSdkVersion="8" android:targetSdkVersion="18" />

<uses-permission android:name="android.permission.VIBRATE" />

<application android:allowBackup="true"

android:icon="@drawable/ic\_launcher" android:label="@string/app\_name"

android:theme="@style/AppTheme" >

<activity android:name="test.alarm.Alaram" android:label="@string/app\_name" >

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<receiver android:name="test.alarm.MyBroadcastReceiver" >

</receiver>

</application>

</manifest>

**MY BROADCAST RECEIVER**

package test.alarm; import android.R;

import android.content.BroadcastReceiver; import android.content.Context;

import android.content.Intent; import android.os.Vibrator; import android.widget.Toast;

public class MyBroadcastReceiver extends BroadcastReceiver {

//MediaPlayer mp; @Override

public void onReceive(Context arg0, Intent arg1) {

//AssetFileDescriptor afd = getAssets().openFd("a1.mp3");

//String mp3File = "raw/a1.mp3";

//File file = null;

//AssetManager assetMan = getAssets();

// TODO Auto-generated method stub

//mp=MediaPlayer.create(arg0, R.)

//..mp.start();

Vibrator vibrator = (Vibrator)arg0.getSystemService(Context.VIBRATOR\_SERVICE);

vibrator.vibrate(2000);

Toast.makeText(arg0, "Alarm. ", Toast.LENGTH\_LONG).show();

}

}

**RESULT:**

Thus the program is executed successfully.

## EX NO: 13 PHOTO

## DATE:

**AIM:**

To develop android application to create photo album.

**ALGORITHM:**

1)Open eclipse or android studio and select new android project 2)Give project name “Familyphoto.java”.and select next.

3) Choose the android version.Choose the lowest android version(Android 2.2) and select next.

4)Enter the package name.package name must be two word seprated by comma and click finish. 5)Go to package explorer in the left hand side.select our project.

6)Go to Graphical Layout to create an image view.

7)Drag the image view icon and click create new icon.

8)Browse your image from your device and click OK, the image icon is created and displayed on your device.

9) Change the height and width of your image.

10)Now run the application so that the family photo is displayed on your android device.



**RESULT:**

Thus the program is executed successfully.

**EX NO: INTRUSION DETECTION SYSTEM**

**DATE:**

**SCOPE OF THE PROJECT:**

This project aims to design and implement a theft detection system using a Passive Infrared (PIR) motion sensor to enhance security and surveillance in homes, offices, and other spaces. Security concerns in residential and commercial areas, along with the limitations of conventional surveillance systems such as high costs and lack of real-time alerts, highlight the need for a cost-effective, efficient, and easily deployable theft detection solution. The system detects motion in a predefined area using a PIR sensor and can optionally integrate advanced features like a camera module for capturing images or video upon detection. Key features include motion detection via infrared radiation changes, real-time alerts for potential theft, scalability through additional sensors, and energy efficiency due to the PIR sensor's low power consumption. Applications span residential security for intrusion detection, commercial security for monitoring offices and warehouses, and perimeter surveillance in sensitive areas like bank vaults and research labs. However, limitations include potential false alarms caused by pets or environmental factors, as well as detection constraints related to line-of-sight, range (typically up to 10 meters), and angle (around 120° field of view). Future enhancements may involve AI or ML algorithms to distinguish between humans and other motion sources, integration of additional sensors such as sound detectors and cameras for comprehensive security, and a buzzer system to alert people nearby.

**COMPONENTS USED:**

a) ESP 8266

b) PIR Motion Sensor

c) Jumper Cables

**WORKING PRINCIPLE:**

The PIR (Passive Infrared) sensor operates by detecting infrared radiation emitted naturally by warm objects such as humans and animals. When any moving object enters its detection range, the sensor identifies a sudden change in IR levels and interprets it as motion detection. This movement causes the PIR sensor to send a digital HIGH signal to the connected microcontroller, such as an Arduino or ESP32, signaling that motion has been detected. If no movement occurs within its sensing area, the PIR sensor maintains a LOW output signal, indicating a calm or idle state. The system primarily remains in this idle state, continuously monitoring the surroundings without any active triggers. However, the moment motion is detected, the system quickly transitions into an active state, which leads to the triggering of alerts. These alerts could include sounding a buzzer, activating a camera to capture images, or sending notifications to authorities. This continuous cycle of monitoring, detecting, and alerting forms the core of a reliable security system, making PIR sensors an essential component in real-time intrusion detection and surveillance setups.

**PROGRAM:**

#define BLYNK\_TEMPLATE\_ID "TMPL3hF72ZMxG"

#define BLYNK\_TEMPLATE\_NAME "Theft Alert using Using 8266"

#define BLYNK\_AUTH\_TOKEN "f3YahsFUNq7bQ92im2MTVGAbYYTueEE7"

#define BLYNK\_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

char auth[] = BLYNK\_AUTH\_TOKEN;

char ssid[] = "######"; // type your wifi name

char pass[] = "x-x-x-x-x-x"; // type your wifi password

#define PIR\_SENSOR 4

BlynkTimer timer;

//int flag=0;

void notifyOnTheft()

{

int isTheftAlert = digitalRead(PIR\_SENSOR);

Serial.println(isTheftAlert);

if (isTheftAlert==1) {

Serial.println("Theft Alert in Home");

// Blynk.email("smadesh1007@gmail.com", "Alert", "Theft Alert in Home");

//Blynk.notify("Alert : Theft Alert in Home");

Blynk.logEvent("theft\_alert","Theft Alert in Home");

// flag=1;

}

else if (isTheftAlert==0)

{

// flag=0;

}

}

void setup(){

pinMode(PIR\_SENSOR, INPUT);

Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

//dht.begin();

timer.setInterval(5000L, notifyOnTheft);

}

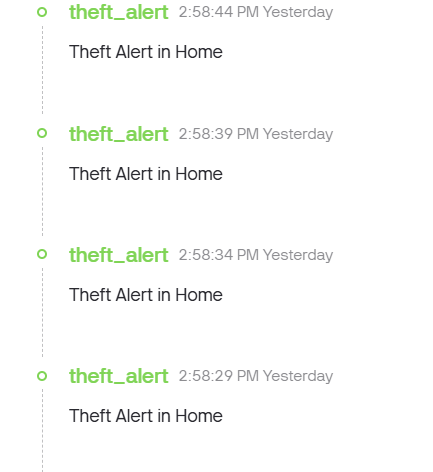
void loop(){

Blynk.run();

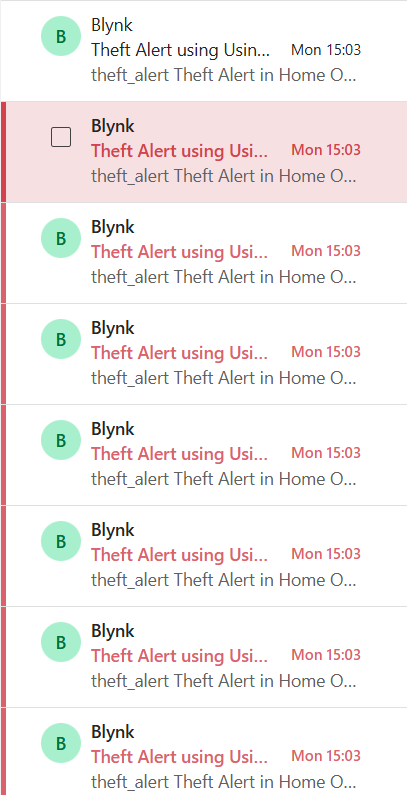
timer.run();

}

**OUTPUT:**

****

****



**RESULT:** Thus Intrusion Detection System has been implemented successfully.