



## **Unit- I - Introduction and UI interface**

- 1. Introduction to mobile technologies**
- 2. Mobile operating systems**
- 3. Mobile devices – pros and cons**
- 4. Introduction to Android,  
Versions,Features**
- 5. Android architecture**
- 6. UI Layouts**
- 7. UI Controls / Widgets**
- 8. Event handling**
- 9. Required Tools- Eclipse, ADT, AVD**
- 10. Application structure**
- 11. Android manifest file**
- 12. Android design philosophy**
- 13. Creating android applications**



## 1. Mobile Networks / Technologies

- ❖ GSM
- ❖ GPRS
- ❖ EDGE
- ❖ 1G, 2G, 3G, 4G, 5G
- ❖ IEEE 802.11
- ❖ Infrared
- ❖ Bluetooth



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



**GSM** – Global System for Mobile Communication.

**GRPS** – General Packet Radio Services.

**EDGE**- Enhanced Data GSM Environment

Faster version the Global System for Mobile (GSM) wireless service designed to deliver data at rates up to 384 Kbps and enable the delivery of multimedia and other broadband applications to mobile phone and computer users.

**IEEE 802.11** - IEEE 802.11 is part of the IEEE 802 set of local area network (LAN) protocols, and specifies the set of media access control (MAC) and physical layer (PHY) protocols for implementing wireless local area network (WLAN).

**Infrared**- infrared network provides a flexible infrastructure for research into wireless mobile computing.

**Bluetooth**- Bluetooth is a low-cost, short distance wireless communications technology, mainly is used to provide communicate capacity between mobile devices and other devices



## Cellular Network

- A cellular network or mobile network is a communication network where the last link is wireless.
- The network is distributed over land areas called "cells", each served by at least one fixed-location transceiver, known as a cell site or base station.
- These base stations provide the cell with the network coverage which can be used for transmission of voice, data, and other types of content.
- A cell typically uses a different set of frequencies from neighbouring cells, to avoid interference and provide guaranteed service quality within each cell.



## Cellular Network Towers



- When joined together these cells provide radio coverage over a wide geographic area.
- This enables a large number of portable transceivers (e.g., mobile phones, pagers, etc.) to communicate with each other and with fixed transceivers and telephones anywhere in the network, via base stations.
- Cellular networks use lower power, shorter range and more transmitters for data transmission.
- Wireless Cellular Systems solves the problem of spectral congestion and increases user capacity.



## **Structure of the mobile phone cellular network:**

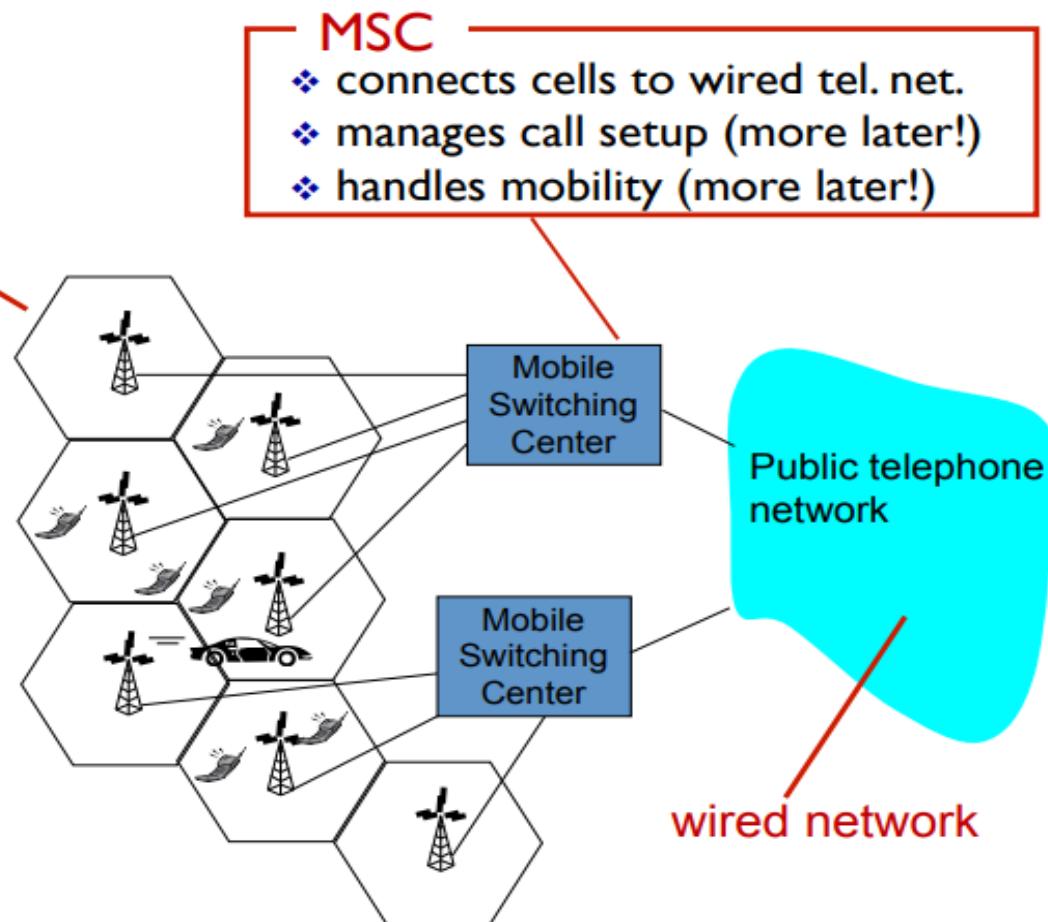
- A simple view of the cellular mobile-radio network consists of the following:
- A network of radio base station forming the base station subsystem
- The core circuit switched network for handling voice calls and text
- A packet switched network for handling mobile data
- The public switched telephone network to connect subscribers to the wider telephony network
- This network is the foundation of the GSM system network. There are many functions that are performed by this network in order to make sure customers get the desired service including mobility management, registration, call set-up, and handover



# Components of Cellular Network

**cell**

- ❖ covers geographical region
- ❖ **base station (BS)** analogous to 802.11 AP
- ❖ **mobile users** attach to network through BS
- ❖ **air-interface:** physical and link layer protocol between mobile and BS





## The functions of MSC are,

- Call set-up and release.
- Routing of calls and messages sent via SMS.
- Managing conference calls and calls on hold.
- Fax services.
- Billing
- Interfacing with other networks like public switched telephone network (PSTN) and Internet

Base stations transmit to and receive from mobiles at the assigned spectrum

- Multiple base stations use the same spectrum (spectral reuse)

The service area of each base station is called a cell. Each mobile terminal is typically served by the closest base stations.



## **Advantages of Cellular Network:**

- It provides voice/data services even while roaming.
- It connects both fixed and wireless telephone users.
- It is used in areas where cables can not be laid out due to its wireless nature.
- It is easy to maintain.
- It is easy to upgrade the equipment's.

## **Disadvantages of Cellular Network:**

- It offers less data rate compare to wired networks such as fiber optics, DSL(Digital Subscriber Line)etc.
- Macro cells are affected by multipath signal loss.
- It requires higher cost in order to setup cellular network infrastructure.



## **Additional features of cellular systems are,**

- Offer very high capacity in a limited spectrum.
- Reuse of radio channel in different cells.
- Enable a fixed number of channels to serve an arbitrarily large number of users by reusing the channel throughout the coverage region.
- Communication is always between mobile and base station (not directly between mobiles).
- Each cellular base station is allocated a group of radio channels within a small geographic area called a cell.
- Neighboring cells are assigned different channel groups.
- By limiting the coverage area to within the boundary of the cell, the channel groups may be reused to cover different cells.
- Keep interference levels within tolerable limits.
- Frequency reuse or frequency planning.
- Organization of Wireless Cellular Network.
- Cellular network is organized into multiple low power transmitters.



## Shape of Cells

- The coverage area of cellular networks are divided into **cells**, each cell having its own antenna for transmitting the signals. Each cell has its own frequencies.
- Data communication in cellular networks is served by its base station transmitter, receiver and its control unit.

The shape of cells can be either square or hexagon

**Square:** A square cell has four neighbors at distance **d** and four at distance Root 2 d  
Better if all adjacent antennas equidistant

Simplifies choosing and switching to new antenna

**Hexagon:** A hexagon cell shape is highly recommended for its easy coverage and calculations.  
It offers the following advantages :

- Provides equidistant antennas.
- Distance from center to vertex equals length of side.



## Frequency Reuse:

- Frequency reusing is the concept of using the same radio frequencies within a given area, that are separated by considerable distance, with minimal interference, to establish communication. Frequency reuse offers the following benefits :
- Allows communications within cell on a given frequency
- Limits escaping power to adjacent cells
- Allows re-use of frequencies in nearby cells
- Uses same frequency for multiple conversations 10 to 50 frequencies per cell
- For example, when **N** cells are using the same number of frequencies and **K** be the total number of frequencies used in systems. Then each **cell frequency** is calculated by using the formula **K/N**.
- In Advanced Mobile Phone Services (AMPS) when  $K = 395$  and  $N = 7$ , then frequencies per cell on an average will be  $395/7 = 56$ . Here, **cell frequency** is 56.



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



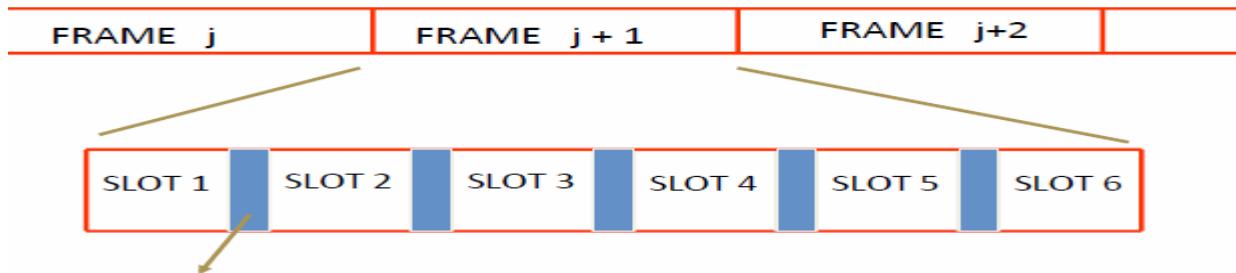
There are 2 techniques for sharing mobile-to-base station radio spectrum are:

### 1. Combined FDMA/TDMA:

It divides spectrum in frequency channel and divides each channel into time slots.

### 2. Code Division Multiple Access (CDMA):

It allows reuse of same spectrum over all cells. Net capacity improvement. Two frequency bands are used one of which is for forward channel (cell-site to subscriber) and one for reverse channel (sub to cell-site).



Guard time – signal transmitted by mobile terminals at different locations do no arrive at the base station at the same time.

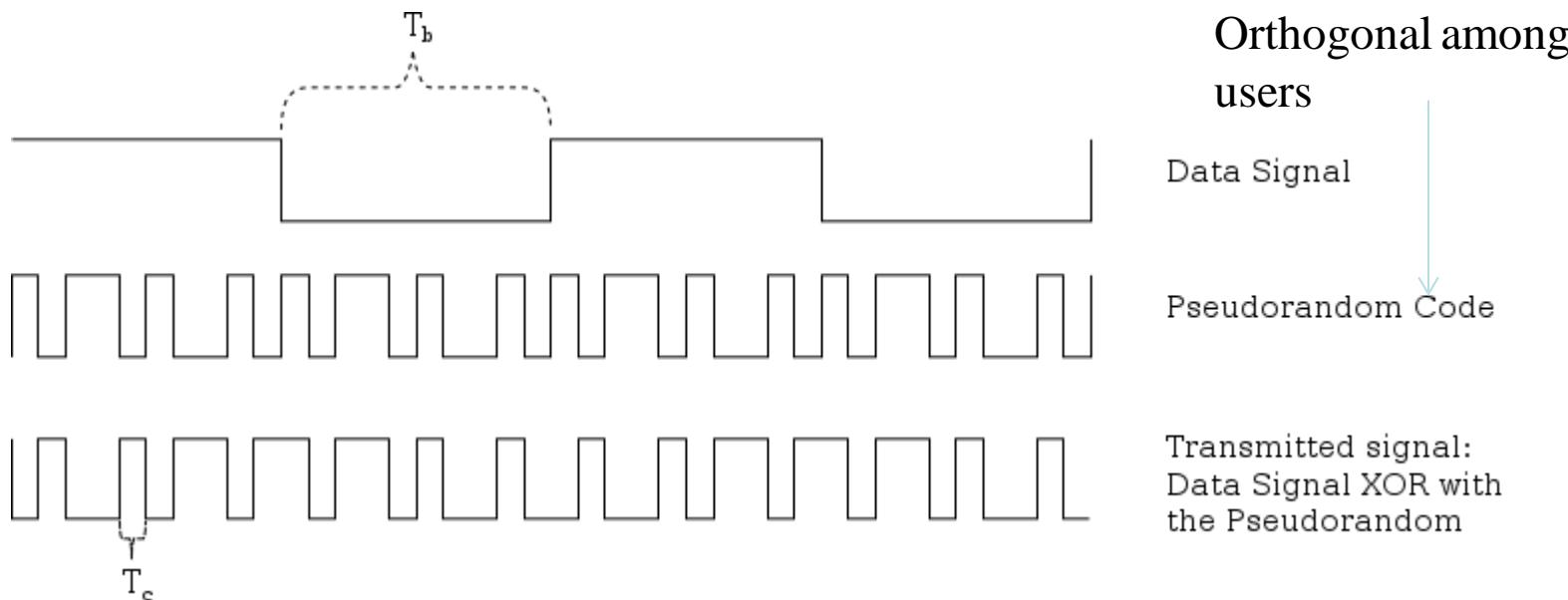
- Time is divided into slots and only one mobile terminal transmits during each slot.

Each user is given a specific slot



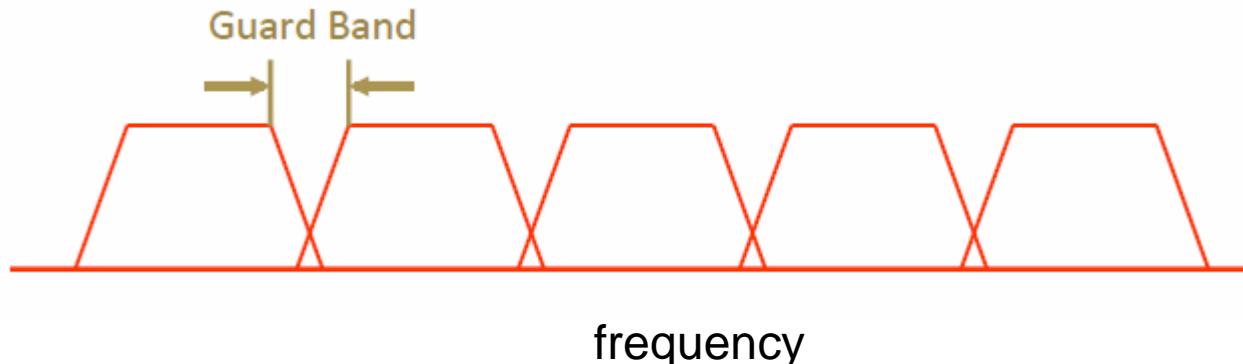
## Code Division Multiple Access

- Use of **orthogonal codes** to separate different transmissions
- Each symbol of bit is transmitted as a larger number of bits using the **user specific code** – Spreading
  - **Bandwidth** occupied by the signal is much larger than the information transmission rate
  - But all users use the **same frequency band** together





## Frequency Division Multiple Access



- Each mobile is assigned a separate frequency channel for the duration of the call
- Sufficient guard band is required to prevent adjacent channel interference
- Usually, mobile terminals will have one downlink frequency band and one uplink frequency band
- Different cellular network protocols use different frequencies



## Cellular Network Generations

- **1G:** Analog cellular telephony.
- **2G:** Digital cellular telephony.
- **3G:** High-speed digital cellular telephony (including video telephony).
- **4G:** IP-based “anytime, anywhere” voice, data, and multimedia telephony at faster data rates than 3G (to be deployed in 2012–2015).



## 1 GENERATION

- First generation cellular networks
- Radio signals = analog
- First generation mobile networks were reliant upon analog radio systems which meant that users could only make phone calls, they couldn't send or receive text messages.



1G TECHNOLOGY

- ★ *1G refers to the first generation of wireless telephone technology, mobile telecommunications which was first introduced in 1980s and completed in early 1990s.*
- ★ *It's Speed was upto 2.4 kbps.*
- ★ *It allows the voice calls in 1 country.*
- ★ *1G network use Analog Signal.*
- ★ *AMPS was first launched in USA in 1G mobile systems.*





**1G**  
FIRST  
GENERATION

### *DRAWBACKS OF 1G*

- ★ *Poor Voice Quality*
- ★ *Poor Battery Life*
- ★ *Large Phone Size*
- ★ *No Security*
- ★ *Limited Capacity*
- ★ *Poor Handoff Reliability*





## 2G (GSM and GPRS Networks)

- 2G carriers continued to improve transmission quality and coverage paging, faxes, text messages and voicemail.
- 2.5G uses GPRS(General Packet Radio Services), which delivers packet-switched capabilities to existing GSM networks.
- The 1G network was not perfect, but it remained until 1991 when it was replaced with 2G. This new mobile network ran on digital signal, not analog, which vastly improved its security but also its capacity. On 2G, users could send SMS and MMS messages (although slowly and often without success) and when GPRS was introduced in 1997, users could receive and send emails on the move.



## 2G TECHNOLOGY

- ❖ *2G technology refers to the 2<sup>nd</sup> generation which is based on GSM.*
- ❖ *It was launched in Finland in the year 1991.*
- ❖ *2G network use digital signals.*
- ❖ *It's data speed was upto 64kbps.*

### Features Includes:

- ✓ *It enables services such as text messages, picture messages and MMS (multi media message).*
- ✓ *It provides better quality and capacity .*





## DRAWBACKS OF 2G

- 2G requires strong digital signals to help mobile phones work. If there is no network coverage in any specific area , digital signals would weak.*
  
- These systems are unable to handle complex data such as Videos.*





## FEATURES OF 3G TECHNOLOGY

- ✓ *Providing Faster Communication*
- ✓ *Send/Receive Large Email Messages*
- ✓ *High Speed Web / More Security*

*Video Conferencing / 3D Gaming*

- ✓ *TV Streaming/ Mobile TV/ Phone Calls*
- ✓ *Large Capacities and Broadband Capabilities*
- ✓ *11 sec – 1.5 min. time to download a 3 min Mp3 song.*





## DRAWBACKS OF 3G TECHNOLOGY

- ◆ *Expensive fees for 3G Licenses Services*
- ◆ *It was challenge to build the infrastructure for 3G*
- ◆ *High Bandwidth Requirement*
- ◆ *Expensive 3G Phones.*
- ◆ *Large Cell Phones*





## 4G TECHNOLOGY (*Anytime ,Anywhere*)

- ◆ 4G technology refer to or short name of fourth Generation which was started from late 2000s.
- ◆ Capable of providing 100Mbps – 1Gbps speed.
- ◆ One of the basic term used to describe 4G is MAGIC.

### **MAGIC:**

- ◆ *Mobile Multimedia*
- ◆ *Anytime Anywhere*
- ◆ *Global Mobility Support*
- ◆ *Integrated Wireless Solution*
- ◆ *Customized Personal Services*

*Also known as Mobile Broadband Everywhere.*

It's five times faster than the 3G network – and can in theory provide speeds of up to 100Mbps. All mobile models released from 2013 onwards should support this network, which can offer connectivity for tablets and laptops as well as smart phones. Under 4G, users can experience better latency (less buffering), higher voice quality, easy access to instant messaging services and social media, quality streaming and make faster downloads.



*4G (Anytime, Anywhere)*

- ◆ *The next generations of wireless technology that promises higher data rates and expanded multimedia services.*
- ◆ *Capable to provide speed 100Mbps-1Gbps.*
- ◆ *High QOS and High Security*
- ◆ *Provide any kind of service at any time as per user requirements, anywhere.*

**Features Include:**

- *More Security*
- *High Speed*
- *High Capacity*
- *Low Cost Per-bit etc.*





## 5G TECHNOLOGY

- ◆ *5G technology refer to short name of fifth Generation which was started from late 2010s.*
- ◆ *Complete wireless communication with almost no limitations.*
- ◆ *It is highly supportable to WWW (Wireless World Wide Web).*





## GSM

- GSM is a globally accepted standard for digital cellular communications.
- GSM uses narrowband Time Division Multiple Access (TDMA) for providing Voice and text based services over mobile phone networks.

### What is GSM?

- If you are in Europe or Asia and using a mobile phone, then most probably you are using GSM technology in your mobile phone.
- GSM stands for **G**lobal **S**ystem for **M**obile **C**ommunication. It is a digital cellular technology used for transmitting mobile voice and data services.
- The concept of GSM emerged from a cell-based mobile radio system at Bell Laboratories in the early 1970s.
- GSM is the name of a standardization group established in 1982 to create a common European mobile telephone standard.
- GSM is the most widely accepted standard in telecommunications and it is implemented



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



- GSM is a circuit-switched system that divides each 200 kHz channel into eight 25 kHz time-slots. GSM operates on the mobile communication bands 900 MHz and 1800 MHz in most parts of the world. In the US, GSM operates in the bands 850 MHz and 1900 MHz. GSM owns a market share of more than 70 percent of the world's digital cellular subscribers. GSM makes use of narrowband Time Division Multiple Access (TDMA) technique for transmitting signals.
- GSM was developed using digital technology. It has an ability to carry 64 kbps to 120 Mbps of data rates..Presently GSM supports more than one billion mobile subscribers in more than 210 countries throughout the world.
- GSM provides basic to advanced voice and data services including roaming service. Roaming is the ability to use your GSM phone number in another GSM network.
- GSM digitizes and compresses data, then sends it down through a channel with two other streams of user data, each in its own timeslot.



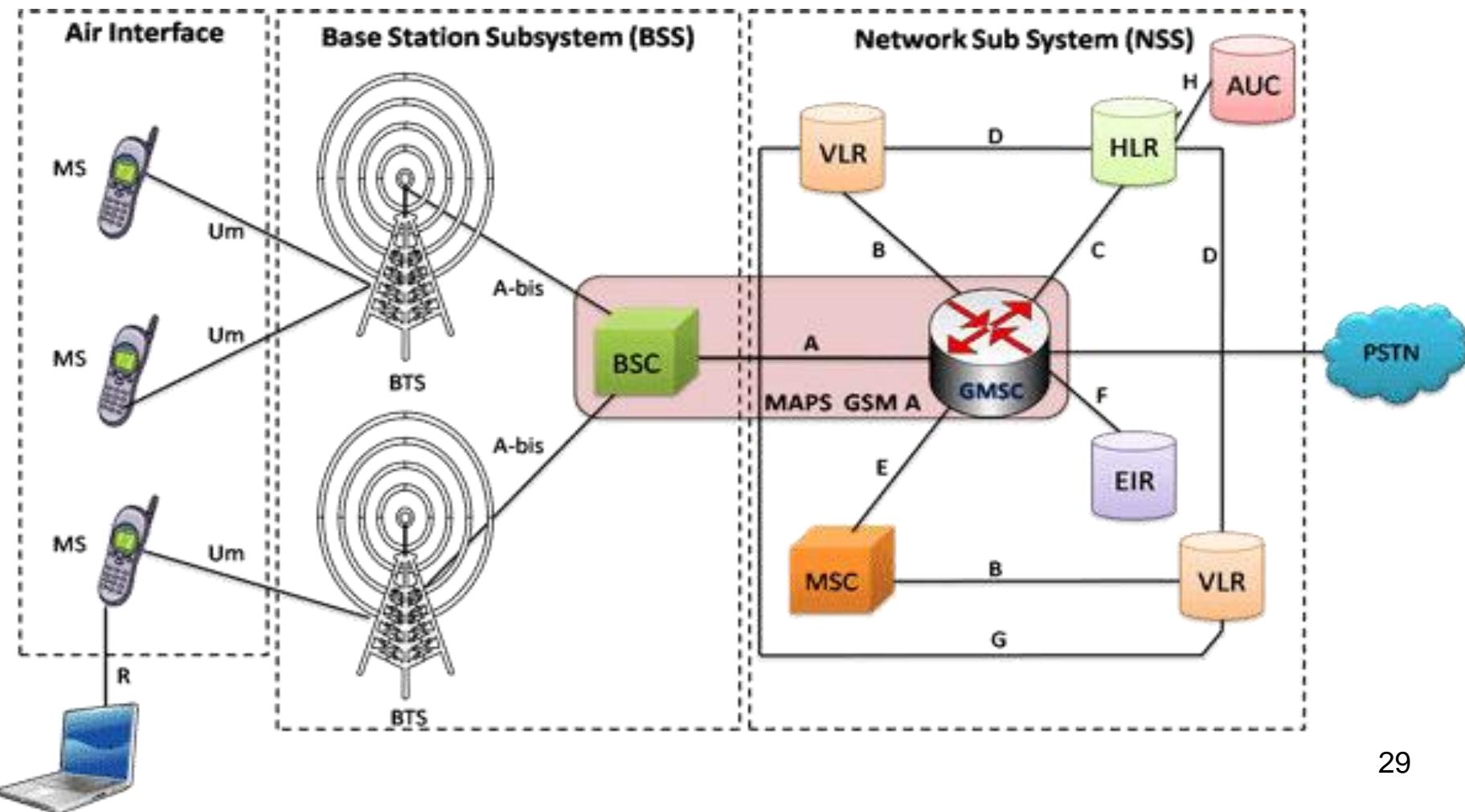
# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



## GSM Architecture

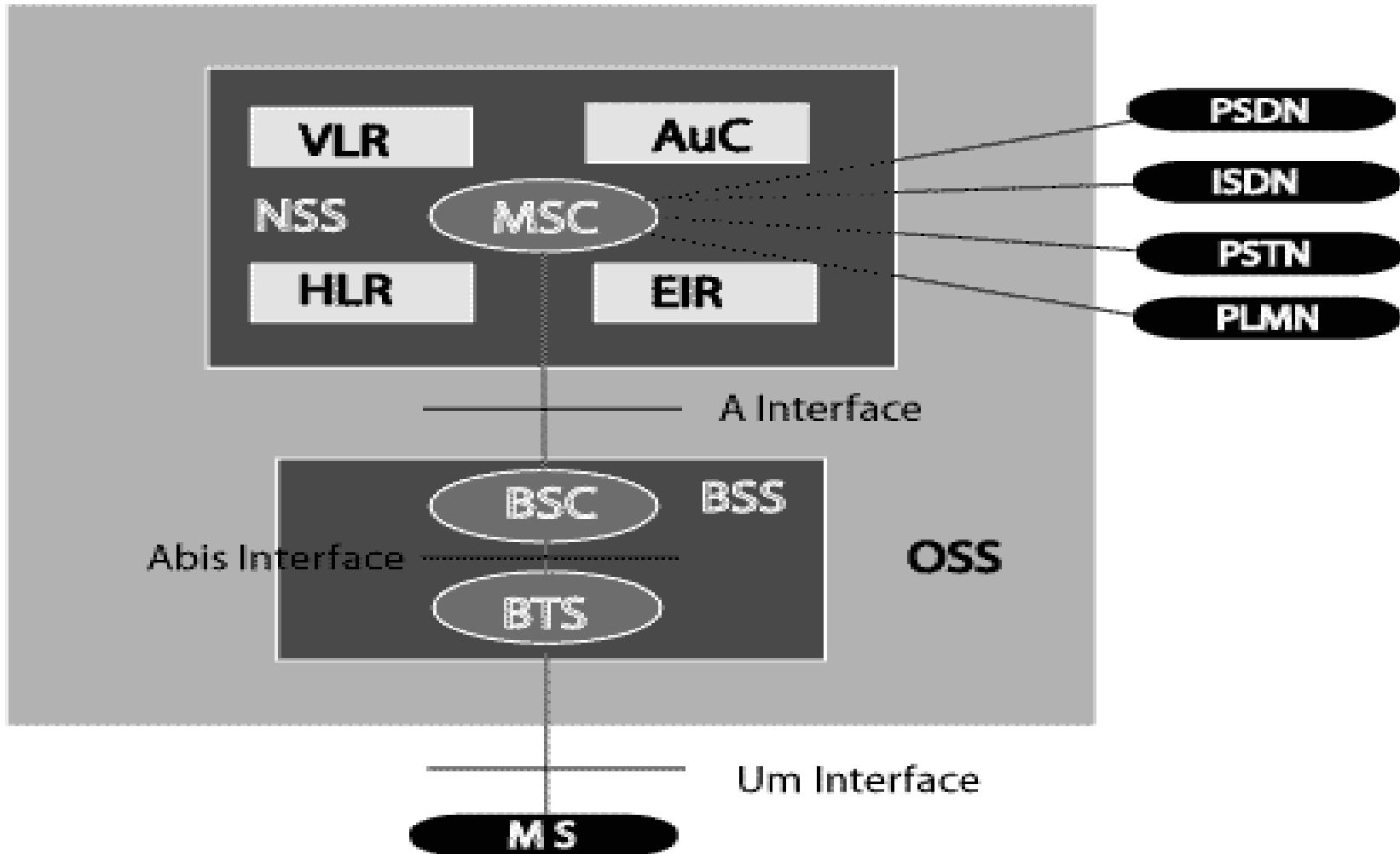




# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC

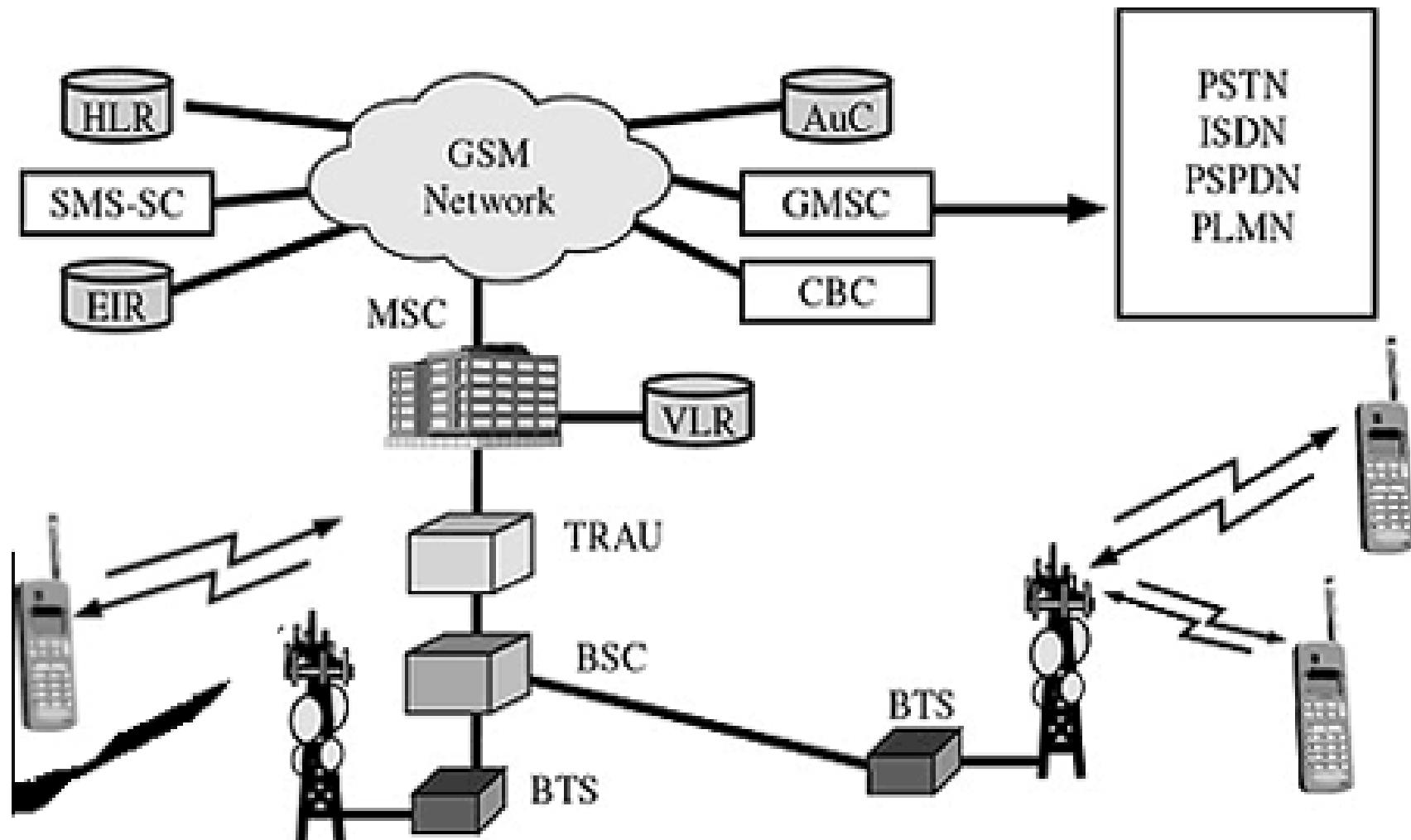




# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



The additional components of the GSM architecture comprise of databases and messaging systems functions:

- Home Location Register (HLR)
- Visitor Location Register (VLR)
- Equipment Identity Register (EIR)
- Authentication Center (AuC)
- SMS Serving Center (SMS SC)
- Gateway MSC (GMSC)
- Chargeback Center (CBC)
- Transcoder and Adaptation Unit (TRAU)



A GSM network comprises of many functional units. The GSM network can be broadly divided into:

1. The Mobile Station (MS)
2. The Base Station Subsystem (BSS)
3. The Network Switching Subsystem (NSS)
4. The Operation Support Subsystem (OSS).

The MS consists of the physical equipment, such as the radio transceiver, display and digital signal processors, and the SIM card. It provides the air interface to the user in GSM networks. As such, other services are also provided, which include:

- Voice teleservices
- Data bearer services
- The features' supplementary service



The BSS is composed of two parts:

1. The Base Transceiver Station (BTS)
2. The Base Station Controller (BSC)

The BTS and the BSC communicate across the specified Abis interface, enabling operations between components that are made by different suppliers. The radio components of a BSS may consist of four to seven or nine cells.

A BSS may have one or more base stations. The BSS uses the Abis interface between the BTS and the BSC. A separate high-speed line (T1 or E1) is then connected from the BSS to the Mobile MSC.



## BTS Functionalities:

- Encoding, encrypting, multiplexing, modulating, and feeding the RF signals to the antenna
- Transcoding and rate adaptation
- Time and frequency synchronizing
- Voice through full- or half-rate services
- Decoding, decrypting, and equalizing received signals
- Random access detection
- Timing advances
- Uplink channel measurement



## **BSC Functionalities:**

- Control of frequency hopping
- Performing traffic concentration to reduce the number of lines from the MSC
- Providing an interface to the Operations and Maintenance Center for the BSS
- Reallocation of frequencies among BTSs
- Time and frequency synchronization
- Power management
- Time-delay measurements of received signals from the MS



## Characteristics of the GSM specification

- **Modulation**

Modulation is the process of transforming the input data into a suitable format for the transmission medium. The transmitted data is demodulated back to its original form at the receiving end. The GSM uses Gaussian Minimum Shift Keying (GMSK) modulation method.

- **Access Methods**

Radio spectrum being a limited resource that is consumed and divided among all the users, GSM devised a combination of TDMA/FDMA as the method to **divide the bandwidth among the users**. In this process, the FDMA part divides the frequency of the total 25 MHz bandwidth into 124 carrier frequencies of 200 kHz bandwidth.

Each BS is assigned with one or multiple frequencies, and each of this frequency is divided into eight timeslots using a TDMA scheme. Each of these slots are used for both transmission as well as reception of data. These slots are separated by time so that a mobile unit doesn't transmit and receive data at the same time.



- **Transmission Rate**

The total symbol rate for GSM at 1 bit per symbol in GMSK produces 270.833 K symbols/second. The gross transmission rate of a timeslot is 22.8 Kbps. GSM is a digital system with an over-the-air bit rate of 270 kbps.

- **Frequency Band**

The uplink frequency range specified for GSM is 933 - 960 MHz (basic 900 MHz band only). The downlink frequency band 890 - 915 MHz (basic 900 MHz band only).

- **Duplex Distance**

Duplex distance is the space between the uplink and downlink frequencies. The duplex distance for GSM is 80 MHz, where each channel has two frequencies that are 80 MHz apart.

- **Channel Spacing**

Channel spacing indicates the spacing between adjacent carrier frequencies. For GSM, it is 200 kHz.



## GSM Operations

Once a Mobile Station initiates a call, a series of events takes place. Analyzing these events can give an insight into the operation of the GSM system.

### **Mobile Phone to Public Switched Telephone Network (PSTN)**

When a mobile subscriber makes a call to a PSTN telephone subscriber, the following sequence of events takes place:

The MSC/VLR receives the message of a call request.

The MSC/VLR checks if the mobile station is authorized to access the network. If so, the mobile station is activated. If the mobile station is not authorized, then the service will be denied.

MSC/VLR analyzes the number and initiates a call setup with the PSTN.

MSC/VLR asks the corresponding BSC to allocate a traffic channel (a radio channel and a time slot).

The BSC allocates the traffic channel and passes the information to the mobile station.

The called party answers the call and the conversation takes place.



## PSTN to Mobile Phone

When a PSTN subscriber calls a mobile station, the following sequence of events takes place:

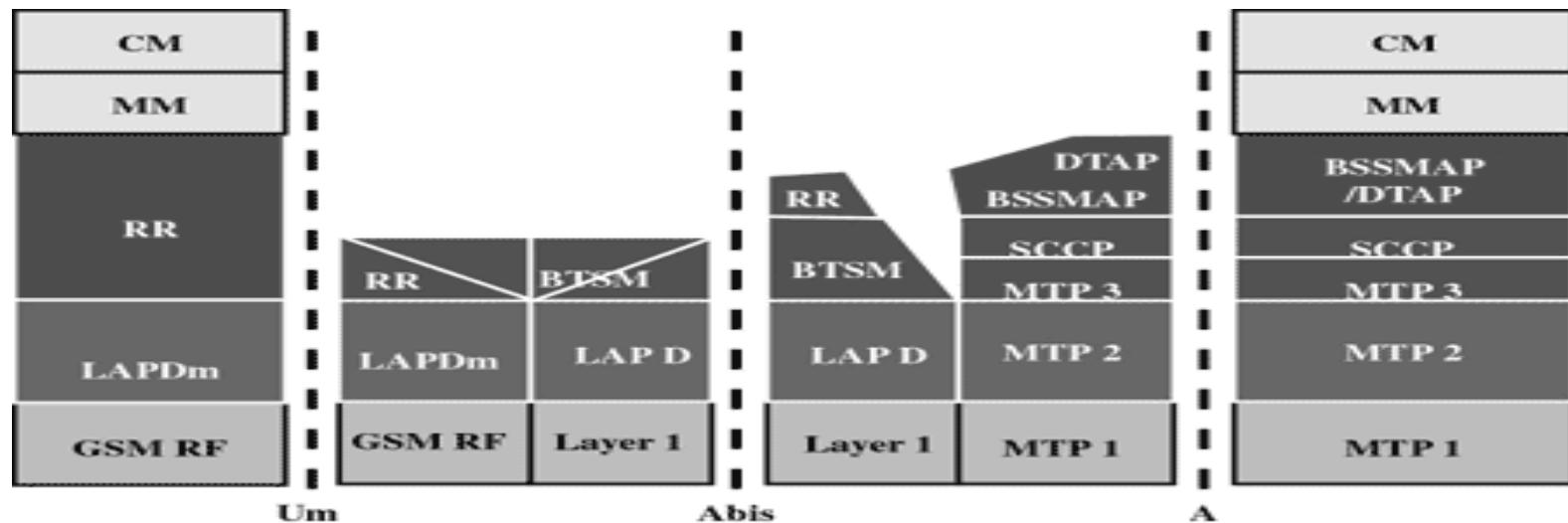
- The Gateway MSC receives the call and queries the HLR for the information needed to route the call to the serving MSC/VLR.
- The GMSC routes the call to the MSC/VLR.
- The MSC checks the VLR for the location area of the MS.
- The MSC contacts the MS via the BSC through a broadcast message, that is, through a paging request.
- The MS responds to the page request.
- The BSC allocates a traffic channel and sends a message to the MS to tune to the channel.
- The MS generates a ringing signal and, after the subscriber answers, the speech connection is established.



## GMS protocol stacks

GSM architecture is a layered model that is designed to allow communications between two different systems. The lower layers assure the services of the upper-layer protocols. Each layer passes suitable notifications to ensure the transmitted data has been formatted, transmitted, and received accurately.

### The GMS protocol stacks diagram





## MS Protocols

Based on the interface, the GSM signaling protocol is assembled into three general layers:

**Layer 1** : The physical layer. It uses the channel structures over the air interface.

**Layer 2** : The data-link layer. Across the Um interface, the data-link layer is a modified version of the Link access protocol for the D channel (LAP-D) protocol used in ISDN, called Link access protocol on the Dm channel (LAP-Dm). Across the A interface, the Message Transfer Part (MTP), Layer 2 of SS7 is used.

**Layer 3** : GSM signalling protocol's third layer is divided into three sublayers:  
Radio Resource Management (RR),  
Mobility Management (MM), and  
Connection Management (CM).



## GSM offers three basic types of services:

1. Telephony services or teleservices
2. Data services or bearer services
3. Supplementary services

### **1. Teleservices**

The abilities of a Bearer Service are used by a Teleservice to transport data. These services are further transited in the following ways:

#### **Voice Calls**

The most basic Teleservice supported by GSM is telephony. This includes full-rate speech at 13 kbps and emergency calls, where the nearest emergency-service provider is notified by dialing three digits.

#### **Videotext and Facsmile**

Another group of teleservices includes Videotext access, Teletex transmission, Facsmile alternate speech and Facsmile Group 3, Automatic Facsmile Group, 3 etc.

#### **Short Text Messages**

Short Messaging Service (SMS) service is a text messaging service that allows sending and receiving text messages on your GSM mobile phone. In addition to simple text messages, other text data including news, sports, financial, language, and location-based data can also be transmitted.



## **Bearer Services**

Data services or Bearer Services are used through a GSM phone. to receive and send data is the essential building block leading to widespread mobile Internet access and mobile data transfer. GSM currently has a data transfer rate of 9.6k. New developments that will push up data transfer rates for GSM users are HSCSD (high speed circuit switched data) and GPRS (general packet radio service) are now available.

## **Supplementary Services**

Supplementary services are additional services that are provided in addition to teleservices and bearer services. These services include caller identification, call forwarding, call waiting, multi-party conversations, and barring of outgoing (international) calls, among others



## GPRS

- General Packet Radio System is also known as GPRS is a third-generation step toward internet access.
- GPRS is also known as GSM-IP that is a Global-System Mobile Communications Internet Protocol as it keeps the users of this system online, allows to make voice calls, and access internet on-the-go.
- Even Time-Division Multiple Access (TDMA) users benefit from this system as it provides packet radio access.
- GPRS also permits the network operators to execute an Internet Protocol (IP) based core architecture for integrated voice and data applications that will continue to be used and expanded for 3G services.
- In the current versions of GPRS, networks based on the Internet Protocol (IP) like the global internet or private/corporate intranets and X.25 networks are supported.
- The GPRS specifications are written by the European Telecommunications Standard Institute (ETSI), the European counterpart of the American National Standard Institute (ANSI).



## Goals of GPRS

GPRS is the first step toward an end-to-end wireless infrastructure and has the following goals:

- i. Open architecture
- ii. Consistent IP services
- iii. Same infrastructure for different air interfaces
- iv. Integrated telephony and Internet infrastructure
- v. Leverage industry investment in IP
- vi. Service innovation independent of infrastructure

## Services Offered:

- i. SMS messaging and broadcasting
- ii. Push-to-talk over cellular
- iii. Instant messaging and presence
- iv. Multimedia messaging service
- v. Point-to-Point and Point-to-Multipoint services

## Protocols supported:

- i. Internet Protocol (IP)
- ii. Point-To-Point Protocol (PPP)



## Characteristics

- **Mobility** - The ability to maintain constant voice and data communications while on the move.
- **Localization** - Allows subscribers to obtain information relevant to their current location.

Using the above three characteristics varied possible applications are being developed to offer to the mobile subscribers. These applications, in general, can be divided into two high-level categories:

- Corporation
- Consumer

These two levels further include:

- **Communications** - E-mail, fax, unified messaging and intranet/internet access, etc.
- **Value-added services** - Information services and games, etc.
- **E-commerce** - Retail, ticket purchasing, banking and financial trading, etc.
- **Location-based applications** - Navigation, traffic conditions, airline/rail schedules and location finder, etc.
- **Advertising** - Advertising may be location sensitive. For example, a user entering a mall can receive advertisements specific to the stores in that mall.



## Benefits Of GPRS:

### i. Mobility:

The capacity to keep up consistent voice and information interchanges while moving.

### ii. Cost Efficient:

Communication via GPRS is cheaper than through the regular GSM network.

### iii. Localization:

Enables customers to acquire data applicable to their present area.

### iv. Easy Billing:

GPRS packet transmission offers an easier to use billing than that offered by circuit switched administrations.



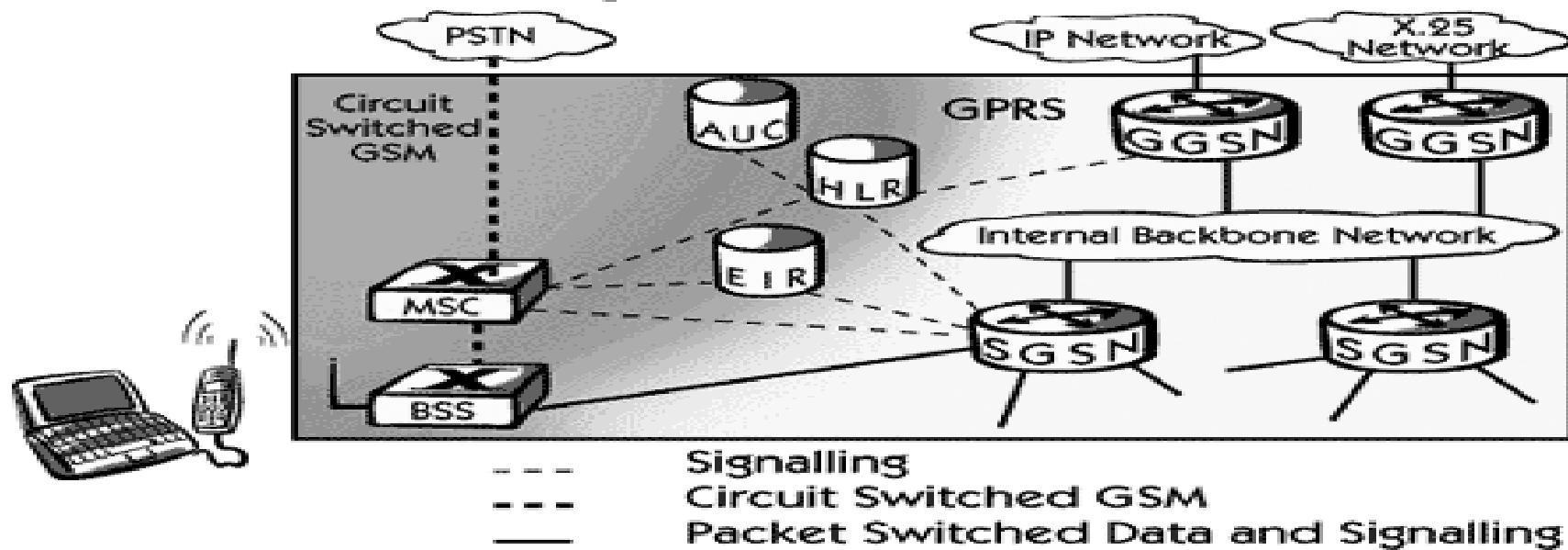
# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



## GPRS architecture

- GPRS architecture works on the same procedure like GSM network, but, has additional entities that allow packet data transmission.
- This data network overlaps a second-generation GSM network providing packet data transport at the rates from 9.6 to 171 kbps.
- Along with the packet data transport the GSM network accommodates multiple users to share the same air interface resources concurrently.





GPRS attempts to reuse the existing GSM network elements as much as possible, but to effectively build a packet-based mobile cellular network, some new network elements, interfaces, and protocols for handling packet traffic are required.

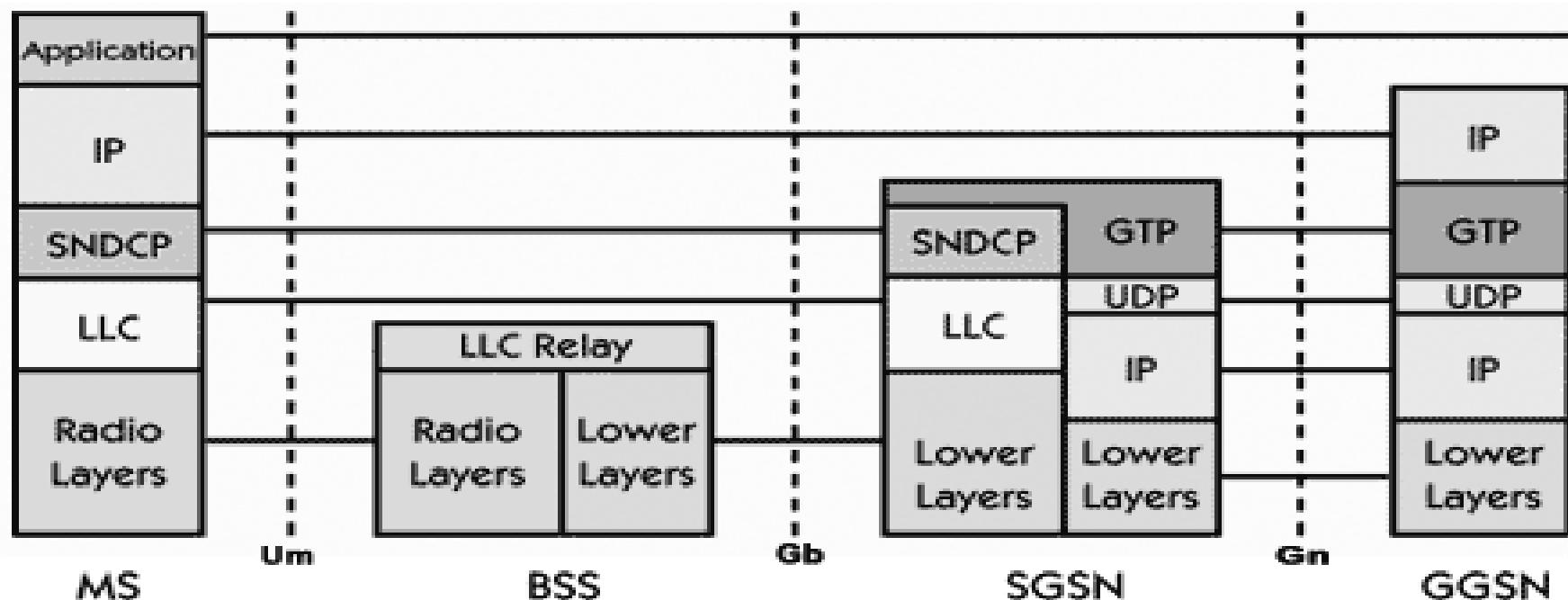
GPRS modifications to numerous GSM network elements are,

<b>GSM Network Element</b>	<b>Modification or Upgrade Required for GPRS.</b>
Mobile Station (MS)	New Mobile Station is required to access GPRS services. These new terminals will be backward compatible with GSM for voice calls.
BTS	A software upgrade is required in the existing Base Transceiver Station(BTS).
BSC	The Base Station Controller (BSC) requires a software upgrade and the installation of new hardware called the packet control unit (PCU). The PCU directs the data traffic to the GPRS network and can be a separate hardware element associated with the BSC.
GPRS Support Nodes (GSNs)	The deployment of GPRS requires the installation of new core network elements called the serving GPRS support node (SGSN) and gateway GPRS support node (GGSN).
Databases (HLR, VLR, etc.)	All the databases involved in the network will require software upgrades to handle the new call models and functions introduced by GPRS.



## GPRS protocol stack

The flow of GPRS protocol stack and end-to-end message from MS to the GGSN is displayed in the below diagram. GTP is the protocol used between the SGSN and GGSN using the Gn interface. This is a Layer 3 tunneling protocol.





## Quality of Service (QoS)

Quality of Service (QoS) requirements of conventional mobile packet data applications are in assorted forms. The QoS is a vital feature of GPRS services as there are different QoS support requirements for assorted GPRS applications like realtime multimedia, web browsing, and e-mail transfer.

GPRS allows defining QoS profiles using the following parameters :

- Service Precedence
- Reliability
- Delay and
- Throughput
- **Service Precedence**

The preference given to a service when compared to another service is known as Service Precedence. This level of priority is classified into three levels called:

- High
- Normal
- Low

When there is network congestion, the packets of low priority are discarded as compared to high or normal priority packets.



- **Reliability**

This parameter signifies the transmission characteristics required by an application. The reliability classes are defined which guarantee certain maximum values for the probability of loss, duplication, mis-sequencing, and corruption of packets.

- **Delay**

The delay is defined as the end-to-end transfer time between two communicating mobile stations or between a mobile station and the GI interface to an external packet data network.

This includes all delays within the GPRS network, e.g., the delay for request and assignment of radio resources and the transit delay in the GPRS backbone network. Transfer delays outside the GPRS network, e.g., in external transit networks, are not taken into account.

- **Throughput**

The throughput specifies the maximum/peak bit rate and the mean bit rate. Using these QoS classes, QoS profiles can be negotiated between the mobile user and the network for each session, depending on the QoS demand and the available resources. The billing of the service is then based on the transmitted data volume, the type of service, and the chosen QoS profile.



## Classes of GPRS

There are two different classes of GPRS terminal equipment's:

- **Class A**

Class A terminals can manage both packet data and voice simultaneously. Which means, one needs two transceivers, as the handset has to send or receive data and voice at the same time. This is the main reason why class A terminals are high-priced to manufacture than class B and C terminals.

- **Class B**

Class B terminals do not play the same role like Class A. These terminals can manage either packet data or voice at a time. One can use a single transceiver for both, resulting in the low cost of terminals.

For example, If a user is using the GPRS session (like WAP browsing, file transfer, etc.) then this session is halted if he or she receives a call. This terminal does not allow both the sessions active in one go. This backlog needs rectification thereby giving the user a facility of both receiving a call and maintaining the data session.



## The GPRS service charging can be based on the following parameters:

- **Volume** - The amount of bytes transferred, i.e., downloaded and uploaded.
- **Duration** - The duration of a PDP context session.
- **Time** - Date, time of day, and day of the week (enabling lower tariffs at offpeak hours).
- **Final destination** - A subscriber could be charged for access to the specific network, such as through a proxy server.
- **Location** - The current location of the subscriber.
- **Quality of Service** - Pay more for higher network priority.
- **SMS** - The SGSN will produce specific CDRs for SMS.
- **Served IMSI/subscriber** - Different subscriber classes (different tariffs for frequent users, businesses, or private users).
- **Reverse charging** - The receiving subscriber is not charged for the received data; instead, the sending party is charged.
- **Free of charge** - Specified data to be free of charge.
- **Flat rate** - A fixed monthly fee.
- **Bearer service** - Charging based on different bearer services



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



### High Speed Circuit Switched Data

Dedicate up to 4 timeslots for data connection  $\sim 50$  kbps Good for real-time applications c.w. GPRS

Inefficient  $\rightarrow$  ties up resources, even when nothing sent

Not as popular as GPRS (many skipping HSCSD)

GSM  
9.6kbps (one timeslot)  
GSM Data  
Also called CSD

HSCSD

GSM

GPRS

WCDMA

Enhanced Data Rates for Global Evolution  
Uses 8PSK modulation  
3x improvement in data rate on short distances  
Can fall back to GMSK for greater distances  
Combine with GPRS (EGPRS)  $\sim 384$  kbps  
Can also be combined with HSCSD

General Packet Radio Services  
Data rates up to  $\sim 115$  kbps  
Max: 8 timeslots used at any one time  
Packet switched; resources not tied up all the time  
Contention based. Efficient, but variable delays  
GSM / GPRS core network re-used by WCDMA(3G)

EDGE



**SATHYABAMA**  
**INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



## 2. Mobile Operating Systems





## What is Mobile OS?

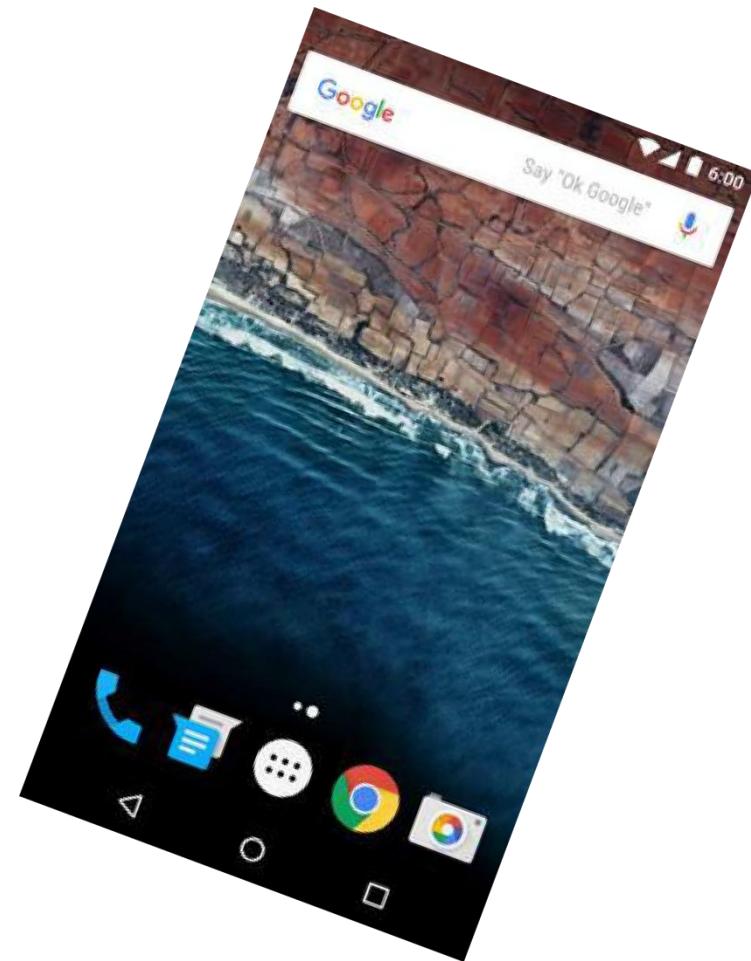
- A Mobile OS is a very basic and essential **software** to operate a Mobile System.
- A Mobile OS is a **software platform** which is designed specially for mobile to handle the devices like Smart phone, Tablet, PDA with lot of features and facilities.
- The operating system is responsible for determining the functions and features available on your device, such as thumb wheel, keyboards, WAP, synchronization with applications, email, text messaging and more.
- The mobile OS will also determine which third-party applications (mobile apps) can be used on your device.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





## Android

- Android is a mobile operating system (OS) currently developed by Google, based on the Linux kernel and designed primarily for touchscreen mobile devices such as smartphones and tablets.
- It was developed by Google, Open Handset Alliance, Android Open Source Project, Android Inc.
- Source model, open source
- Written in C (core), C++, and Java (UI)
- OS family, Unix



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



## Apple iOS





## iOS

-iOS (originally iPhone OS) is a mobile operating system created and developed by Apple Inc. and distributed exclusively for Apple hardware. It is the operating system that presently powers many of the company's mobile devices, including the iPhone, iPad, and iPod touch.

- It was developed by Apple Inc. June 29, 2007
- Source model, closed source
- Written in, C,C++, Objective-C, and Swift
- OS family, Unix





# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Windows





## Windows Mobile

-Windows Mobile is a mobile operating system developed by Microsoft for smart phones and Pocket PC's

-It was first launched in October 2010 with Windows Phone 7

- Currently maintained with Microsoft Corporation
- Written in C, C++
- OS Family, Microsoft Windows



**Windows Phone**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Blackberry



# Blackberry

- BlackBerry OS is a proprietary mobile operating system developed by BlackBerry Ltd for its BlackBerry line of smartphone handheld devices.
  - It was developed by BlackBerry January 19, 1999
  - Source model is **closed source**
  - Written in, **C++ and Java**
  - OS family, **Mobile Operating Systems**

 **BlackBerry**



**SATHYABAMA**  
**INSTITUTE OF SCIENCE AND TECHNOLOGY**  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



**symbian**

# Symbian





## Symbian

- Symbian is a mobile operating system (OS) and computing platform designed for smart phones
- Symbian was originally developed as a closed- source OS for PDAs in 1998 by Symbian Ltd.
- Currently maintained by Accenture on behalf of

Nokia (historically Symbian Ltd. and Symbian Foundation)

- Written in C++
- OS Family RTOS





**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Bada





- Bada is an operating system for mobile devices such as smartphones and tablet computers.
- It was developed by Samsung Electronics on April 2010.
- Source model is, Mixed: proprietary and open source
  - Written in C++
  - OS Family, POSIX (Portable Operating System Interface for Unix)





# Types of Mobile Applications

- Types
  - Web Apps
  - Native Apps
  - Hybrid Apps



## 1. Native Apps:

A native application is a software program that is developed for use on a particular platform or device

- It is **live on the device** and are accessed through icons on the device home screen.
- They are **installed** through an application store (such as Google Play or Apple's App Store).
- They are **developed specifically for one platform**, and can take full advantage of all the device features — they can use the camera, the GPS, the accelerometer, the compass, the list of contacts, and so on.

**Technology Used:** Native apps are coded using a variety of programming languages. Some examples include: Java, Kotlin, Python, Swift, Objective-C, C++, and React.  
Native app = Highest cost, more development time needed (90+ days)



## • Web Apps

- A web application (or web app) is an application software that runs on a web server.
- Web applications are accessed by the user through a web browser with an active internet connection.
- These applications are programmed using a client–server modeled structure.
- Examples of commonly-used, web applications, include: webmail, online retail sales, online banking
- They are **run by a browser** and typically written in HTML5
- Web app = Medium to low cost, can be built in 30-60 days



- Hybrid apps
  - Hybrid apps are **part native apps, part web apps.**
  - Like native apps, they **live in an app store** and can take advantage of the many device features available.
  - Like web apps, they **rely on HTML being rendered in a browser**, with the caveat that the browser is embedded within the app.



### 3. Mobile Devices: Advantages (as compared to fixed devices)

- Always with the user
- Typically have Internet access
- Typically GPS enabled
- Typically have accelerometer & compass
- Mostly have cameras & microphones
- Many apps are free or low-cost and etc...



# Mobile Devices: Limitations

- Limited memory
- Limited processing power
- Different technologies and standards
- Limited or awkward input: soft keyboard, phone keypad, touch screen, or stylus
- Small screens
- Limited and slow network access
- Slow hardware
- Limited battery life
- Limited web browser functionality
- Often inconsistent platforms across devices and etc...



# Android

## Mobile Application Development



## What is 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.
- Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.
- The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.
- On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.
- The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

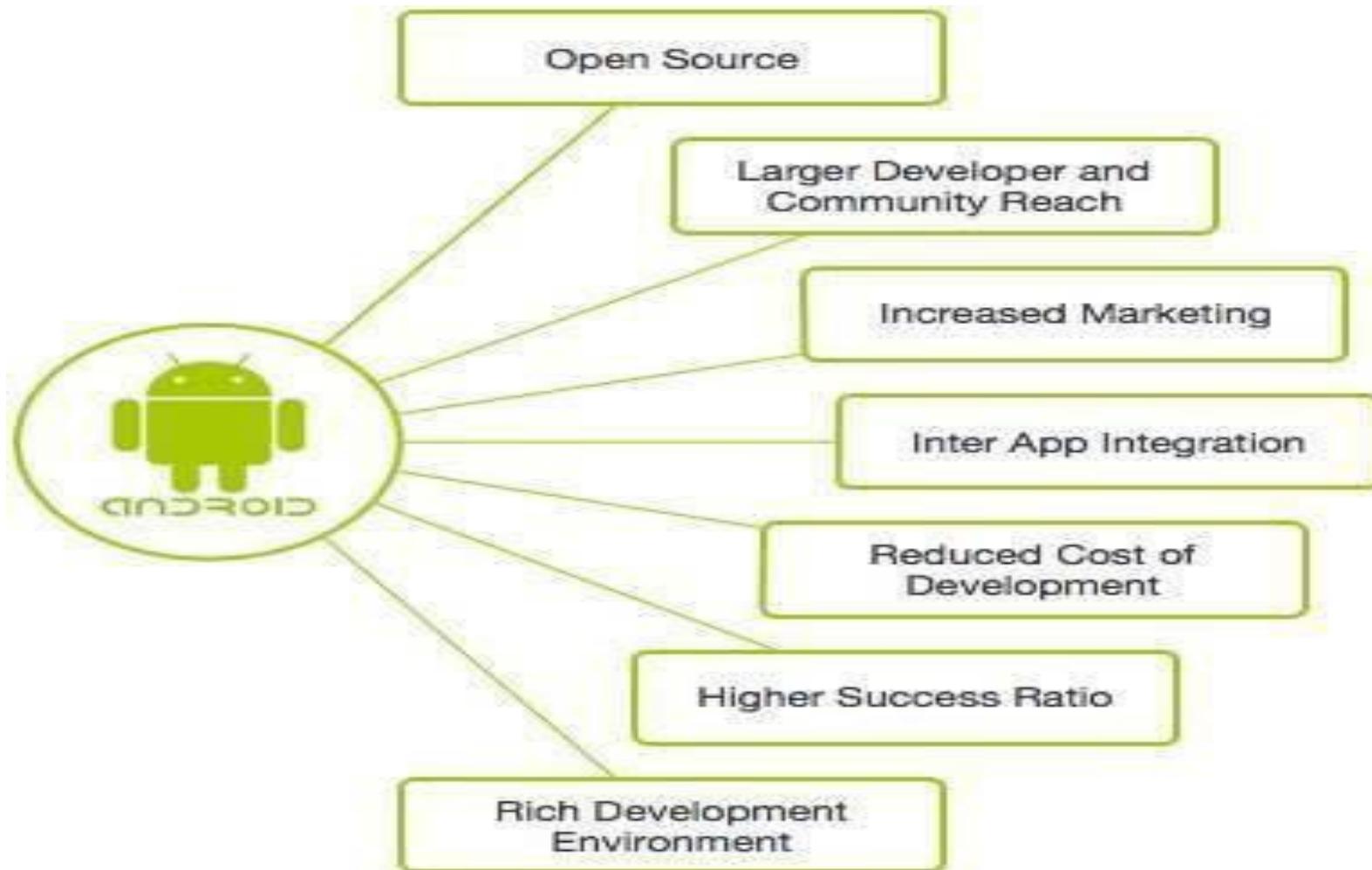


## Features of Android

- Android is a powerful open source operating system which provides immense features and some of these are listed below.
- Android Open Source Project so we can customize the OS based on our requirements.
- Android supports different types of connectivity for GSM, CDMA, Wifi, Bluetooth etc. for telephonic conversation or data transfer.
- Using wifi technology we can pair with other devices while playing games or using other application.
- It contains multiple APIs to support a location-tracking services such as GPS.
- We can manage all data storage related activities by using file manager.
- It contains a wide range of media supports like AVI, MKV, FLV, MPEG4 etc. to play or record variety of audio / video.
- It also support different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
- It supports multimedia hardware control to perform playback or recording using camera and microphone.
- Android has an integrated open source webkit layout based web browser to support User Interface like HTML5, CSS3.
- Android supports a multi-tasking means we can run multiple application at a time and can switch in between them.
- It provide support for virtual reality or 2D/3D Graphics



## Why Android





## Features of Android

- Android is a powerful operating system competing with Apple 4GS and supports great features.

Features	Description
Beautiful UI	Android OS basic screen provides a beautiful and intuitive user interface.
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Storage	SQLite, a lightweight relational database, is used for data storage purposes.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)



Features	Description
Media support	H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP
Messaging	SMS and MMS
Web browser	Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
Multi-tasking	User can jump from one task to another and same time various application can run simultaneously.



Features	Description
Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space
Multi-Language	Supports single direction and bi-directional text.
GCM	Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices.
Wi-Fi Direct	A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
Android Beam	A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.



# Android Components

Application components are the essential building blocks of an Android application.

## Activities

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

## Services

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



## Broadcast Receivers

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

## Content Providers

A content provider component supplies data from one application to others on request.

## Additional Components

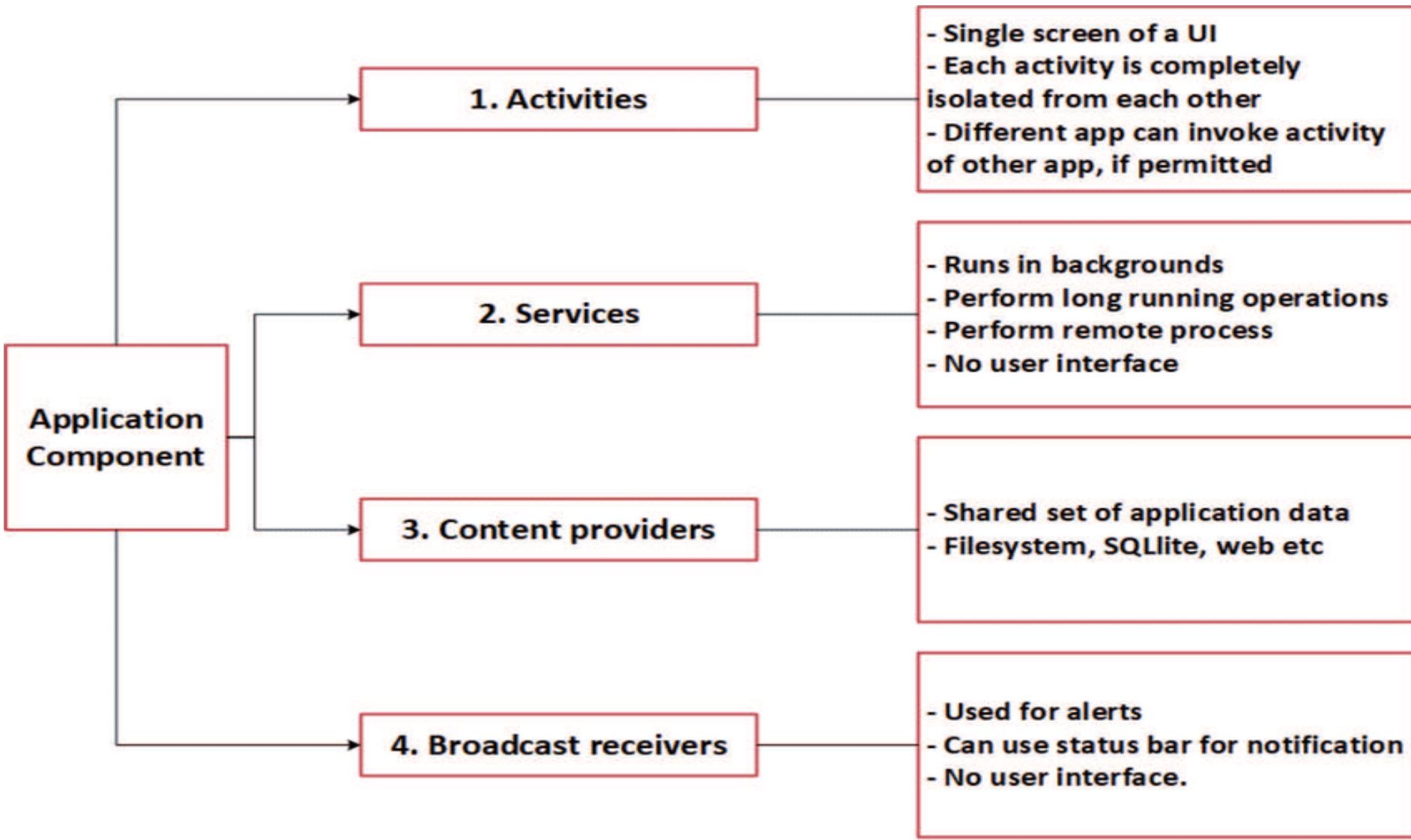
There are additional components which will be used in the construction of above mentioned entities



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





The main components of any android application are the following:

### Activities

- Services
- Content Providers
- Intent and broadcast receivers
- Widgets and Notifications

### 1. Activities

- We can call the Activity as the presentation layer of an Android application. Simply put, an Activity represents the screen on your Android application which has its user interface.
- An application, for instance, an Email App can have many activities such as opening an email, composing an email, replying to an email – these all are different activities. So every Android application has more than one activity.
- When we start a new activity (like replying to an email), previous activity is pushed to the back stack and it gets stopped until the new activity is finished, however, if we push back button while ongoing activity, the current activity gets dissolved and is popped out of the stack and previous activity resumes.



## 2. Services

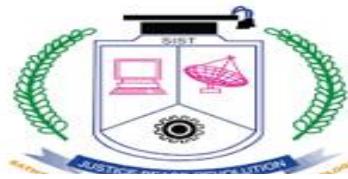
The other important component of an Android application is the service. It performs running operations (long or short) in the background for the activity that you perform on your screen. For example, a push notification from an email. It is possible that service still runs while you have terminated the application or you are not using it currently. For example, when you get an email, you get the notification while still, you are not using the application currently.

## 3. Content Providers

Content Providers manage the application Data and encapsulate it (Object Oriented Feature). This provides the data from one processor of an application to another one. The data might be stored in Database or in a file system or any other storage management systems. Android devices include several native Content Providers that expose useful databases such as the media store and contacts.

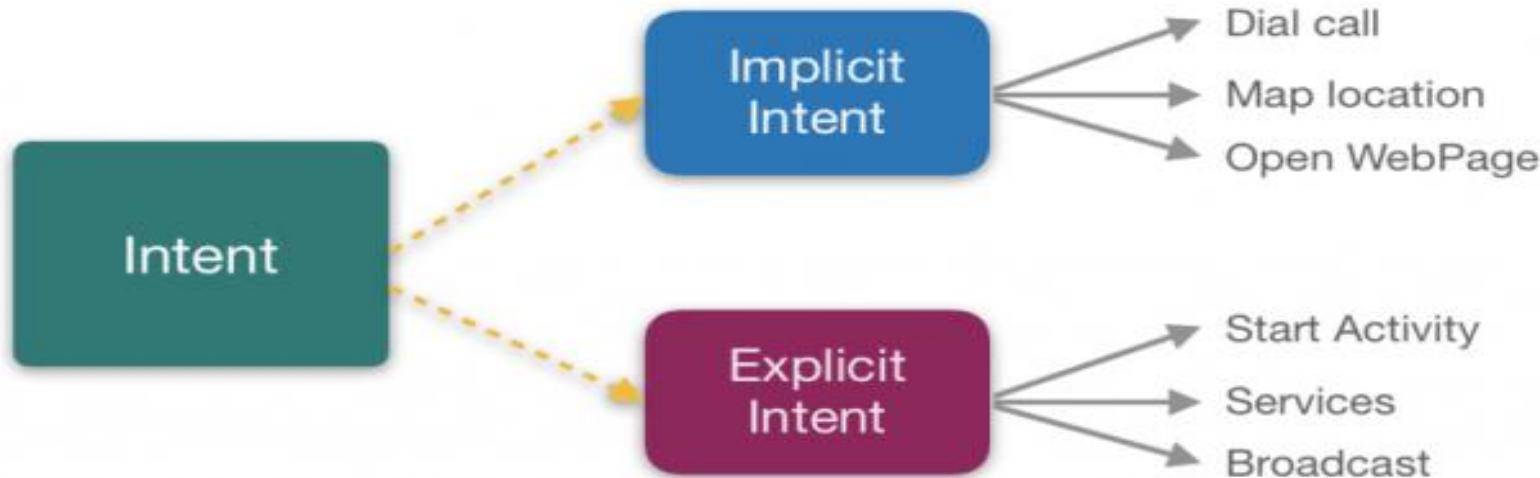
## 4. Intent and Broadcast Receivers

Android Intents are the means of communication that acts as a facilitator when the exchange of message occurs between different components within the same application or from one application to another. In order to start any service, we have to pass an intent to perform this task. Intents are of two types:



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)



**Implicit Intents:** It does not declare the name of the service to start but declares the action to perform.

**Explicit Intents:** It specifies the exact activity to which intent should be given



# SATHYABAMA

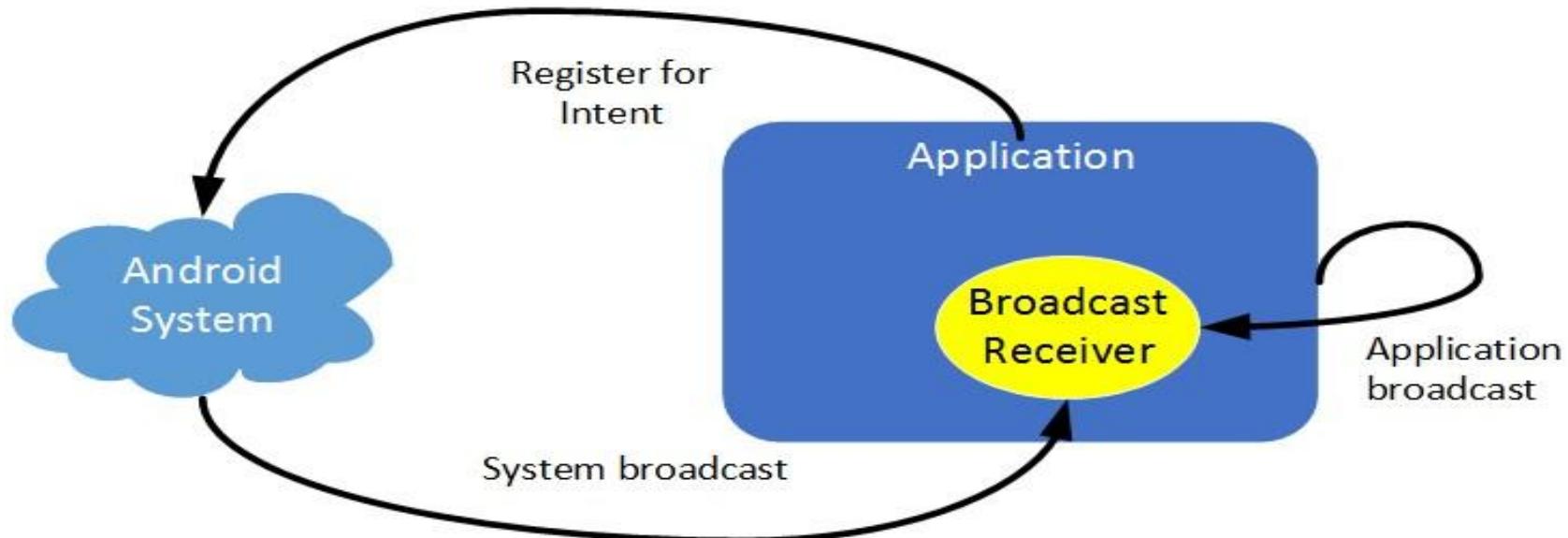
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Broadcast Receivers enable your application to listen for intents that match the criteria you specify. As an example, applications can start the broadcasts to let other applications know that few data has been downloaded to the device and is available for them to use. There are two types of broadcast:

**Normal Broadcast:** They are completely asynchronous and all receivers of the broadcast are run in an undefined order.

**Ordered Broadcast:** They are synchronous and are delivered to one receiver one at a time.





## Categories of Android applications

There are many android applications in the market. The top categories are:

1. Entertainment
2. Tools
3. Communication
4. Productivity
5. Personalization
6. Music and Audio
7. Social
8. Media and Video
9. Travel and Local etc



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



- 
- Android is a software stack for mobile devices that includes an operating system, middleware and key applications.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



- The code names of android ranges from A to N currently, such as
  - 1.0 Astro (some times says no code name)
  - 1.1 Bender (Some times say “Petit four”)
  - 1.5 Cupcake
  - 1.6 Donut
  - 2.x Eclair
  - 2.2 Froyo
  - 2.3.x Gingerbread
  - 3.x.x Honeycomb
  - 4.0.x Ice Cream Sandwitch
  - 4.1.x, 4.2.x and 4.3.x Jelly Bean
  - 4.4.x KitKat and
  - 5.x.x Lollipop
  - 6.0 MarshMallow
  - N (“A Few Weeks”)



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



## Android Versions, Codename and API

Version	Code name	API Level
1.5	Cupcake	3
1.6	Donut	4
2.1	Eclair	7
2.2	Froyo	8
2.3	Gingerbread	9 and 10
3.1 and 3.3	Honeycomb	12 and 13
4.0	Ice Cream Sandwich	15
4.1, 4.2 and 4.3	Jelly Bean	16, 17 and 18
4.4	KitKat	19
5.0	Lollipop	21
6.0	Marshmallow	23
7.0	Nougat	24-25
8.0	Oreo	26-27

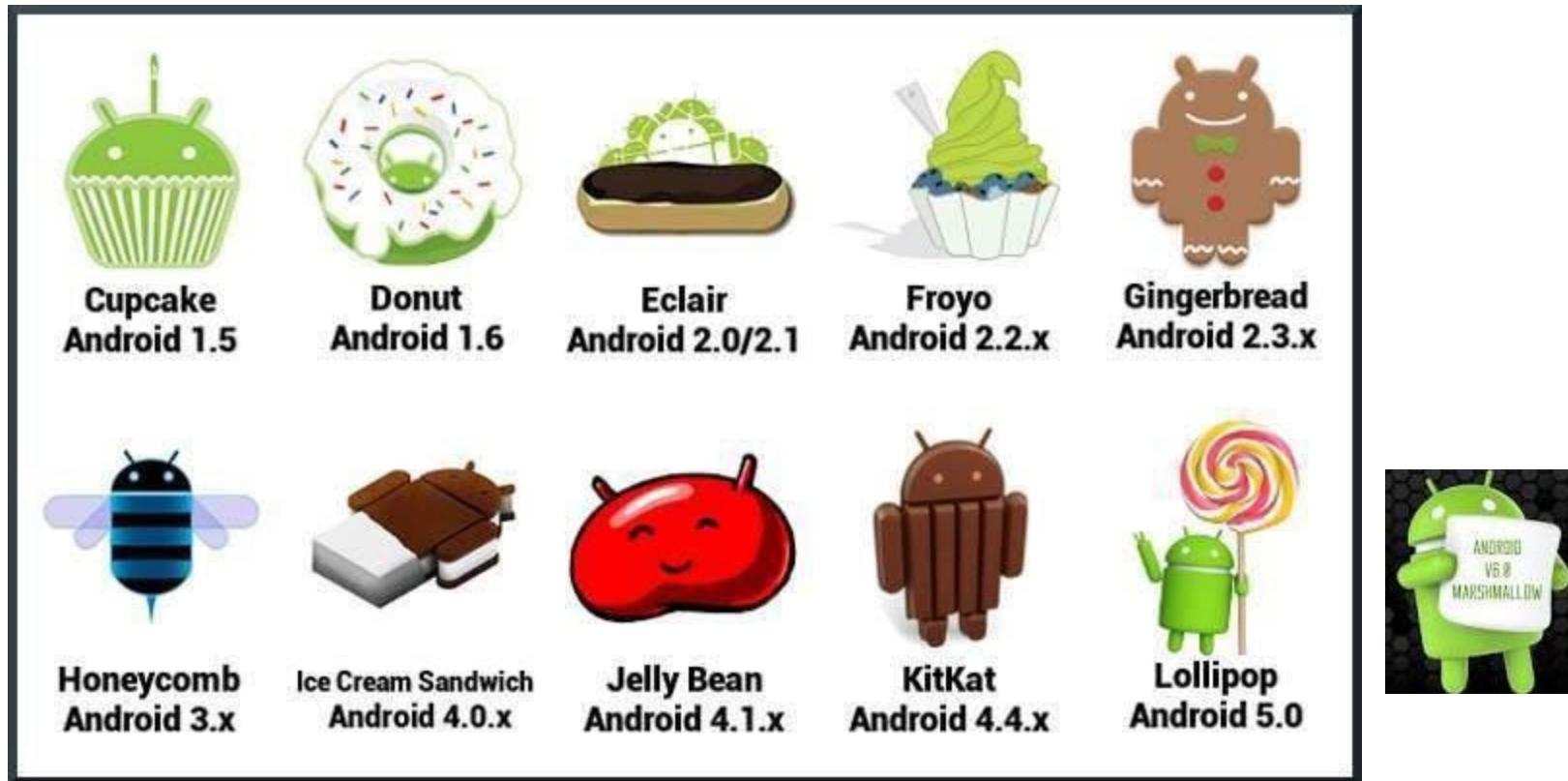


# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



- Let's understand the android history in a sequence.





## Categories of Android applications

- There are many android applications in the market



Music



News



Multimedia



Sports



Lifestyle



Food & Drink



Travel



Weather



Books



Business



Reference



Navigation



Social Media



Utilities



Finance

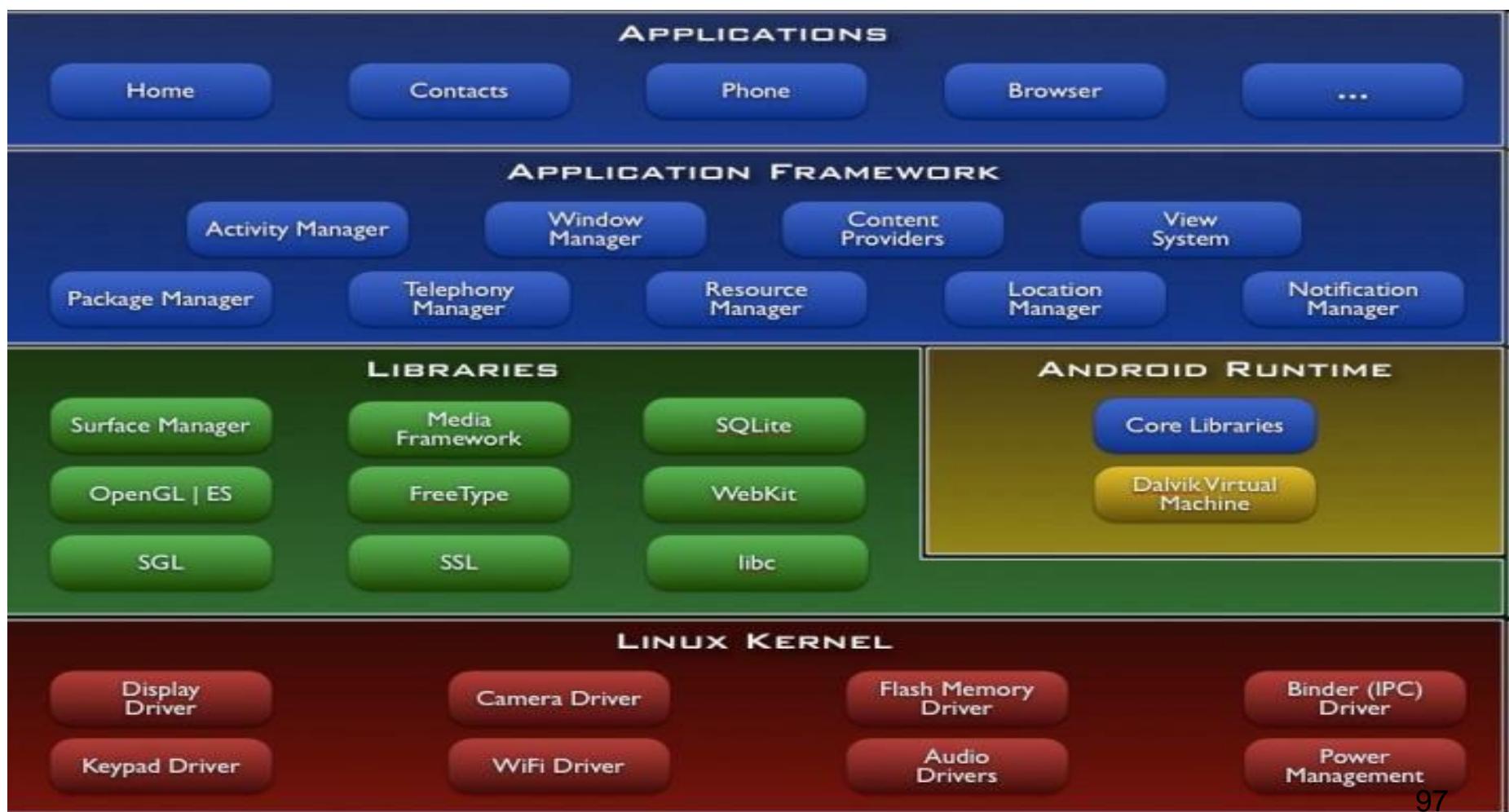


## Android Features

- **Application framework** enabling reuse and replacement of components
- **Dalvik virtual machine** optimized for mobile devices
- **Integrated browser** based on the open source webkit engine
- **Optimized graphics** powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
- **SQLite** for structured data storage
- **Media support** for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)



## 5. Android Architecture

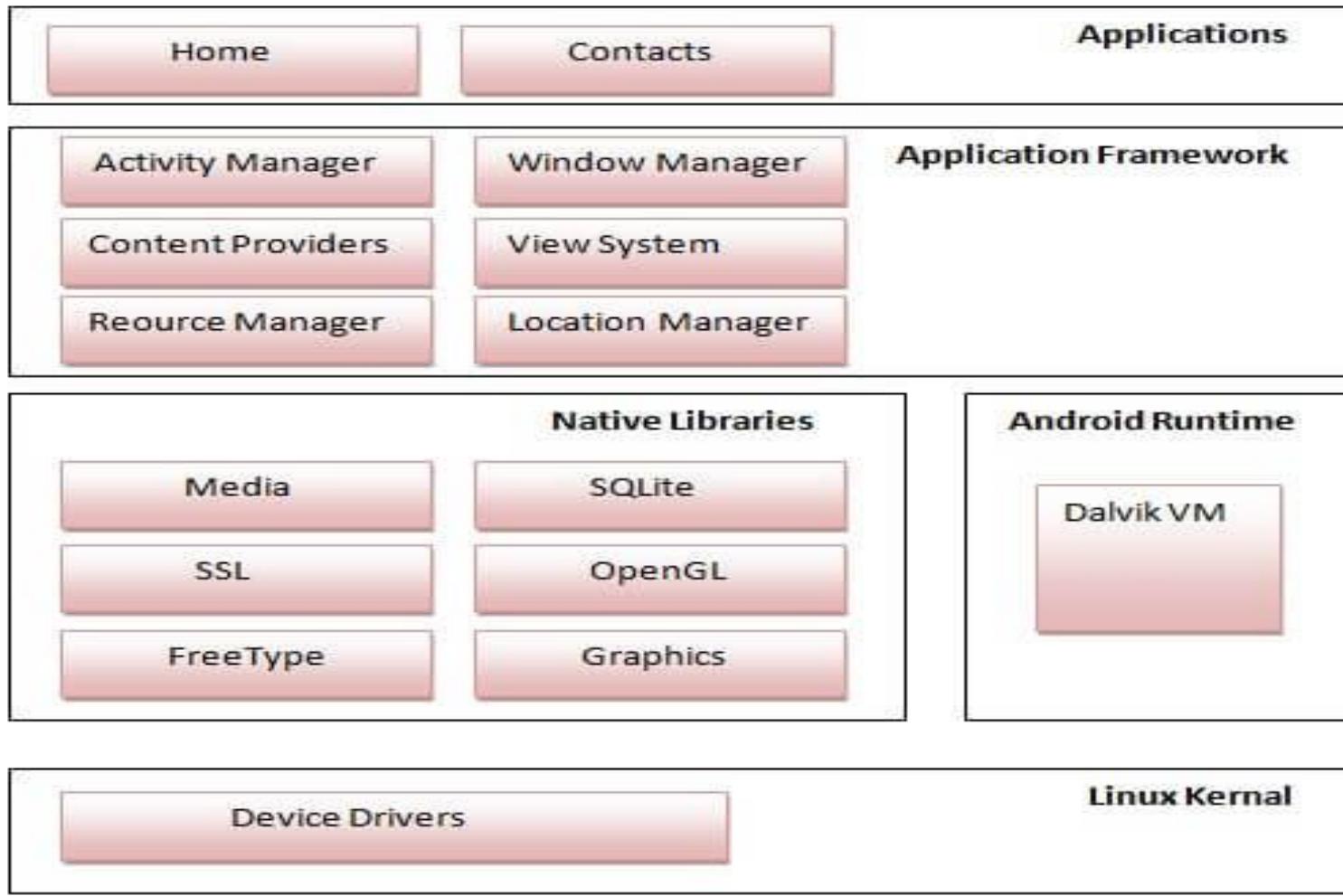




# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





The main components of android architecture are

- Applications
- Application Framework
- Android Runtime
- Platform Libraries
- Linux Kernel
- **Applications**

Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only. It runs within the Android run time with the help of the classes and services provided by the application framework.

- **Application framework**

Application Framework provides several important classes which are used to create an Android application. It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.



Four main components that can be used within an Android application :

Sr.No	Components & Description
1	Activities They dictate the UI and handle the user interaction to the smart phone screen.
2	Services They handle background processing associated with an application.
3	Broadcast Receivers They handle communication between Android OS and applications.
4	Content Providers They handle data and database management issues.



- **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 context, Intent 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 ContentResolver 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 **ContentProvider** class and must implement a standard set of APIs that enable other applications to perform transactions.



```
public class MyContentProvider extends ContentProvider
```

```
{  
    public void onCreate(){ } }
```

S.No	Additional Components & Description
1	<b>Fragments</b> Represents a portion of user interface in an Activity.
2	<b>Views</b> UI elements that are drawn on-screen including buttons, lists forms etc.
3	<b>Layouts</b> View hierarchies that control screen format and appearance of the views.
4	<b>Intents</b> Messages wiring components together.
5	<b>Resources</b> External elements, such as strings, constants and drawable pictures.
6	<b>Manifest</b> Configuration file for the application.



- **Application runtime –**

Android Runtime environment is one of the most important part of Android. It contains components like core libraries and the Dalvik virtual machine(DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries.

Like Java Virtual Machine (JVM), **Dalvik Virtual Machine (DVM)** is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It depends on the layer Linux kernel for threading and low-level memory management. The core libraries enable us to implement android applications using the standard JAVA or Kotlin programming languages.

- **Platform libraries –**

The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.

- i. **Media** library provides support to play and record an audio and video formats.
- ii. **Surface manager** responsible for managing access to the display subsystem.



- iii. **SGL** and **OpenGL** both cross-language, cross-platform application program interface (API) are used for 2D and 3D computer graphics.
- iv. **SQLite** provides database support and **Free Type** provides font support.
- v. **Web-Kit** This open source web browser engine provides all the functionality to display web content and to simplify page loading.
- vi. **SSL (Secure Sockets Layer)** is security technology to establish an encrypted link between a web server and a web browser.

- **Linux Kernel**

Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.

The Linux Kernel will provide an abstraction layer between the device hardware and the other components of android architecture. It is responsible for management of memory, power, devices etc.



The features of Linux kernel are:

**Security:** The Linux kernel handles the security between the application and the system.

**Memory Management:** It efficiently handles the memory management thereby providing the freedom to develop our apps.

**Process Management:** It manages the process well, allocates resources to processes whenever they need them.

**Network Stack:** It effectively handles the network communication.

**Driver Model:** It ensures that the application works properly on the device and hardware manufacturers responsible for building their drivers into the Linux build

- Libraries

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.



- **Android Libraries**

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.

- **android.app** – Provides access to the application model and is the cornerstone of all Android applications.
- **android.content** – Facilitates content access, publishing and messaging between applications and application components.
- **android.database** – Used to access data published by content providers and includes SQLite database management classes.
- **android.opengl** – A Java interface to the OpenGL ES 3D graphics rendering API.
- **android.os** – Provides applications with access to standard operating system services including messages, system services and inter-process communication.
- **android.text** – Used to render and manipulate text on a device display.
- **android.view** – The fundamental building blocks of application user interfaces.
- **android.widget** – A rich collection of pre-built user interface components such as buttons, labels, list views, layout managers, radio buttons etc.
- **android.webkit** – A set of classes intended to allow web-browsing capabilities to be built into applications.



- **Android Runtime**

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called **Dalvik Virtual Machine** which is a kind of Java Virtual Machine specially designed and optimized for Android.

The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine.

The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

- **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 key services are,

**Activity Manager** – Controls all aspects of the application lifecycle and activity stack.

**Content Providers** – Allows applications to publish and share data with other applications.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



**Resource Manager** – Provides access to non-code embedded resources such as strings, color settings and user interface layouts.

**Notifications Manager** – Allows applications to display alerts and notifications to the user.

**View System** – An extensible set of views used to create application user interfaces.



- Android provides a set of core **applications**:
  - ✓ Email Client
  - ✓ SMS Program
  - ✓ Calendar
  - ✓ Maps
  - ✓ Browser
  - ✓ Contacts
  - ✓ Etc
- All applications are written using the **Java language**.



- Enabling and simplifying the **reuse of components**
  - ✓ Developers have full access to the same framework APIs used by the core applications.
  - ✓ Users are allowed to replace components.



## • Features

Feature	Role
View System	Used to build an application, including lists, grids, text boxes, buttons, and embedded web browser
Content Provider	Enabling applications to access data from other applications or to share their own data
Resource Manager	Providing access to non-code resources (localized strings, graphics, and layout files)
Notification Manager	Enabling all applications to display customer alerts in the status bar
Activity Manager	Managing the lifecycle of applications and providing a common navigation back-stack



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



- Including a set of **C/C++ libraries** used by components of the Android system
- Interface through **Java**
- Surface manager – Handling **UI Windows**
- **2D and 3D graphics**
- Media codes, **SQLite**, Browser engine



- Core Libraries

- ✓ Providing most of the functionality available in the core libraries of the Java language

- ✓ APIs

- Data Structures
- Utilities
- File Access
- Network Access
- Graphics





- **Dalvik Virtual Machine**

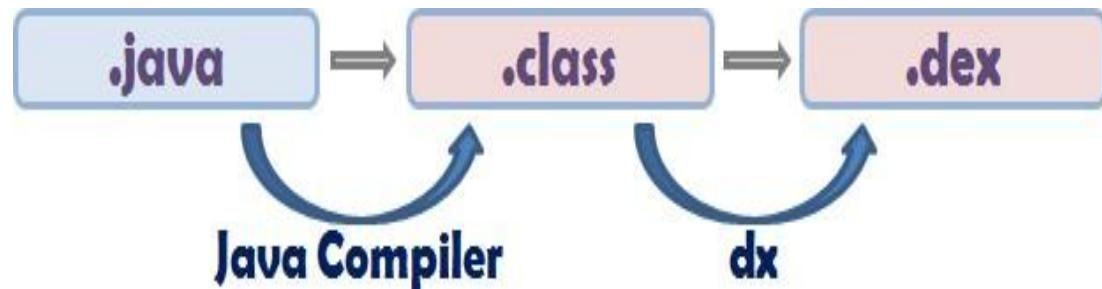
- ✓ Providing environment on which every Android application runs
  - Each Android application runs in its own process, with its own instance of the **Dalvik VM**.
  - Dalvik has been written such that a device can run **multiple VMs efficiently**.
- ✓ Register-based virtual machine



- Dalvik Virtual Machine (Cont)
  - ✓ Executing the Dalvik Executable (.dex) format

➤ .dex format is optimized for minimal memory footprint.

➤ Compilation



- ✓ Relying on the Linux Kernel for:

➤ Threading

➤ Low-level memory management



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



- Relying on **Linux Kernel 2.6** for core system services
  - ✓ Memory and Process Management
  - ✓ Network Stack
  - ✓ Driver Model
  - ✓ Security
- Providing an **abstraction layer** between the H/W and the rest of the S/W stack



- **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.

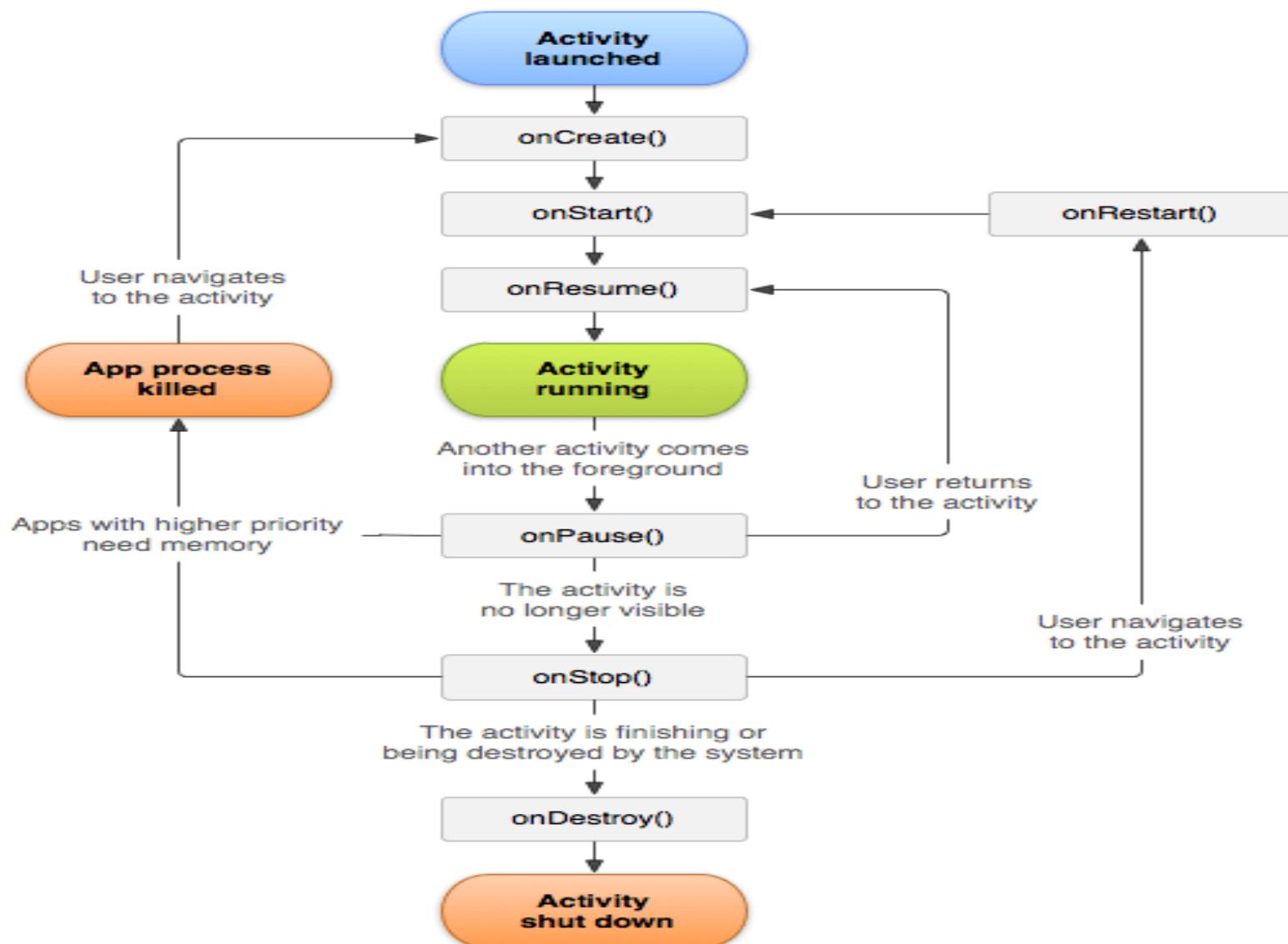
Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed.



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



You must be aware that a program starts from a main() function in different programming languages. Similarly, android initiates the program within an activity with a call to onCreate() callback method. There is a sequence of callback methods that starts up an activity and then tear down in different methods shown in the above Activity life cycle diagram:

- 1. onCreate():** In this state, the activity is created.
  - 2. onStart():** This callback method is called when the activity becomes visible to the user.
  - 3. onResume():** The activity is in the foreground and the user can interact with it.
  - 4. onPause():** Activity is partially obscured by another activity. Another activity that's in the foreground is semi-transparent.
  - 5. onStop():** The activity is completely hidden and not visible to the user.
  - 6. onRestart():** From the Stopped state, the activity either comes back to interact with the user or the activity is finished running and goes away. If the activity comes back, the system invokes onRestart()
  - 7. onDestroy():** Activity is destroyed and removed from the memory.
- So these are the various methods of the Activity Life Cycle.



## Requirement

- The Android emulator takes additional requirements beyond the basic system requirement for Android Studio. These requirements are given below:

SDK Tools 26.1.1 or higher

64-bit processor

Windows: CPU with UG (unrestricted guest) support

HAXM 6.2.1 or later (recommended HAXM 7.2.0 or later)

- **Install the emulator**

- ✓ The Android emulator is installed while installing the Android Studio. However some components of emulator may or may not be installed while installing Android Studio.
- ✓ To install the emulator component, select the Android Emulator component in the SDK Tools tab of the SDK Manager.
- ✓ Run an Android app on the Emulator
- ✓ We can run an Android app from the Android Studio project, or we can run an app which is installed on the Android Emulator as we run any app on a device.
- ✓ To start the Android Emulator and run an application in our project:

1. In Android Studio, we need to create an Android Virtual Device (AVD) that the emulator can use to install and run your app. To create a new AVD:-
  - 1.1 Open the AVD Manager by clicking Tools > AVD Manager.



- 1.2. Click on Create Virtual Device, at the bottom of the AVD Manager dialog. Then Select Hardware page appears.
- 1.3. Select a hardware profile and then click Next. The System Image page appears.
- 1.4. Select the system image for the particular API level and click Next. This leads to open a Verify Configuration page.
- 1.5. Change AVD properties if needed, and then click Finish.
2. In the toolbar, choose the AVD, which we want to run our app from the target device from the drop-down menu
3. Click Run.

### **Launch the Emulator without first running an app**

- ✓ To start the emulator:
- ✓ Open the AVD Manager.
- ✓ Double-click an AVD, or click **Run**
- ✓ While the emulator is running, we can run the Android Studio project and select the emulator as the target device. We can also drag an APKs file to install on an emulator, and then run them.
- ✓ Start the emulator from the command line
- ✓ The Android SDK includes the Android device emulator. Android emulator lets you develop and test out the application without using a physical device.



## Install Android

- Android supports java, c++, c# etc. language to develop android applications. Java is the officially supported language for android.
- All the android examples of this site is developed using Java language and Eclipse IDE.
- The required softwares to develop android applications using Eclipse IDE.

**There are two ways to install android.**

1. By ADT Bundle
2. By Setup Eclipse Manually

### **1) By Android Studio**

It is the simplest technique to install required software for android application. It includes:

- ✓ Eclipse IDE
- ✓ Android SDK
- ✓ Eclipse Plugin

If you download the Android Studio from android site, you don't need to have eclipse IDE, android SDK and eclipse Plugin because it is already included in Android Studio



## 2) By set up eclipse manually.

- Install the JDK
- Download and install the Eclipse for developing android application
- Download and Install the android SDK
- Intall the ADT plugin for eclipse
- Configure the ADT plugin
- Create the AVD

### Create the hello android application.

#### 1) Install the Java Development Kit (JDK)

- For creating android application, JDK must be installed if you are developing the android application with Java language.

#### 2) Download and install the Eclipse IDE

- For developing the android application using eclipse IDE, you need to install the Eclipse.

#### 3) Download and install the android SDK

First of all, download the android SDK. In this example we have installed the android SDK for windows (.exe version).

Now double click on the exe file, it will be installed. I am using the android 2.2 version here.



#### **4) Download the ADT plugin for eclipse**

- ADT (Android Development Tools) is required for developing the android application in the eclipse IDE. It is the plugin for Eclipse IDE that is designed to provide the integrated environment.

**For downloading the ADT, you need to follow these steps:**

- Start the eclipse IDE, then select Help > Install new software...
- In the work with combo box, write <https://dl-ssl.google.com/android/eclipse/>
- select the checkbox next to Developer Tools and click next
- A list of tools to be downloaded here, click next
- click finish

After completing the installation, restart the eclipse IDE.

#### **5) Configuring the ADT plugin**

After the installing ADT plugin, now tell the eclipse IDE for your android SDK location.

Select the Window menu > preferences

Now select the android from the left panel. Here you may see a dialog box asking if you want to send the statistics to the google. Click proceed.

Click on the browse button and locate your SDK directory e.g. my SDK location is C:\Program Files\Android\android-sdk .

Click the apply button then OK.



## 6) Create an Android Virtual Device (AVD)

- For running the android application in the Android Emulator, you need to create and AVD. For creating the AVD:
  - Select the **Window menu > AVD Manager**
  - Click on the **new** button, to create the AVD
  - Now a dialog appears, write the AVD name e.g. myavd.
  - Now choose the target android version e.g. android2.2.
  - click the **create AVD**.

## 7) Create and run the simple android example

- Create the new android project
- Write the message (optional)
- Run the android application



## Android development setup

Follow the instructions ...

Download the software from the URL:

<http://developer.android.com/sdk/index.html>

Install the following Softwares:

- Android SDK
- Eclipse IDE (3.4 or newer)
- Android Development Tools (ADT) Eclipse plug-in

Bring with you (optional):

- Android OS enabled Mobile device
- USB cable so you can test your app on your phone



## Application Fundamentals

- Apps are written in Java
- Bundled by Android Asset Packaging Tool
- Every App runs its own Linux process
- Each process has it's own Java Virtual Machine
- Each App is assigned a unique Linux user ID
- Apps can share the same user ID to see each other's files



## Applications

- Lifestyle applications for senior citizens
- Environmental applications that give data about pollution levels.
- Emergency services ( Hospitals, Police station etc.,)
- Bus services
- Games
- E-governance
- Google map

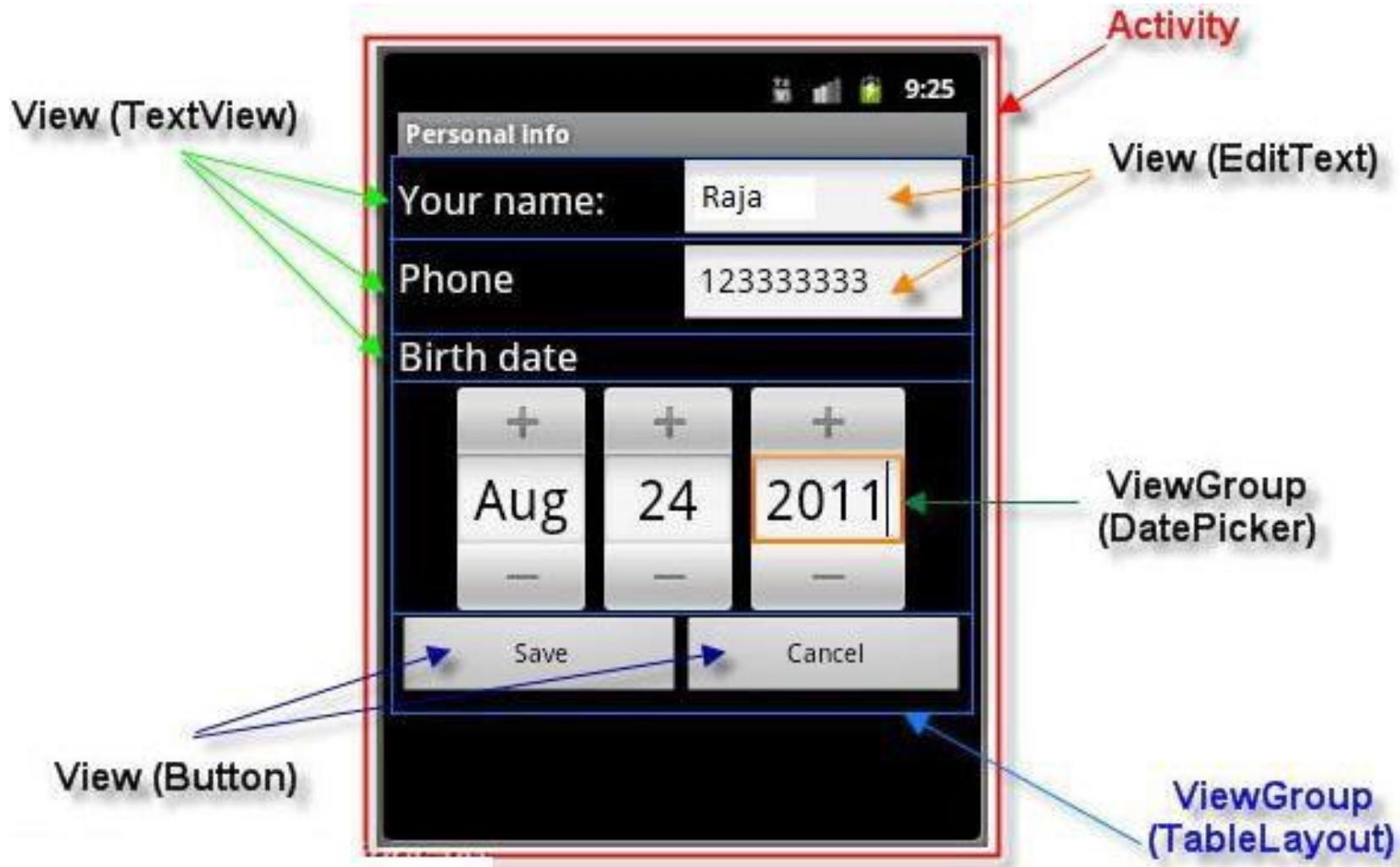


## 6. UI Layouts

- The basic building block for user interface is a **View object** which is created from the **View class**
- It occupies a **rectangular area** on the screen and is responsible for **drawing** and **event handling**.
- View is the base class for widgets, which are used **to create interactive UI components** like buttons, text fields, etc.



- The **ViewGroup** is a subclass of **View** and provides invisible container that hold other **Views** or other **ViewGroups** and define their layout properties.
- At third level we have **different layouts** which are subclasses of **ViewGroup** class
- A typical **layout defines the visual structure** for an Android user interface.





- To declare the layout using simple XML file **main\_layout.xml** which is located in the **res/layout** folder of your project.
- A **layout** may contain any type of widgets such as buttons, labels, textboxes, and so on.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="This is a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:text="This is a Button" />
    <!-- More GUI components go here -->
</LinearLayout>
```



- Once the layout has created, it can **loaded by the help of application code**

- Sample Code

```
public void onCreate(Bundle  
 savedInstanceState)
```

```
{
```

```
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
}
```



# Layout Types

- Linear Layout
- Relative Layout
- Table Layout
- Absolute Layout
- Frame Layout
- List View
- Grid View



# Linear Layout

- Linear Layout is a view group that aligns all children in either **vertically or horizontally**.



LINEAR LAYOUT



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:gravity	This specifies how an object should position its content, on both the X and Y axes. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.
android:orientation	This specifies the direction of arrangement and you will use "horizontal" for a row, "vertical" for a column. The default is horizontal.



# Example

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

```
<LinearLayout
```

```
    xmlns:android=http://schemas.android.com/apk/res/android"
```

```
        android:layout_width="fill_parent"
```

```
        android:layout_height="fill_parent"
```

```
        android:orientation="vertical" >
```

```
<!-- More GUI components go here -->
```

```
</LinearLayout>
```



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output

demo

START_SERVICE
PAUSE_SERVICE
STOP_SERVICE

demo

START_SERVICE	PAUSE_SERVICE	STOP_SERVICE
---------------	---------------	--------------



# Relative Layout

- Relative Layout enables you to specify how **child views** are positioned relative to each other.
- The position of each view can be specified as **relative to sibling elements** or **relative to the parent**.





Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:gravity	This specifies how an object should position its content, on both the X and Y axes. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.  By default, all child views are drawn at the top-left of the layout, so you must define the position of each view using the various layout properties.



# Example

## <RelativeLayout

```
xmlns:android="http://schemas.android.co  
m/apk/res/android"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:paddingLeft="16dp"  
    android:paddingRight="16dp" >
```

```
<!-- More GUI components go here -->  
</RelativeLayout>
```



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output

demo

---

Enter your name

---

NEW BUTTON

---

NEW BUTTON



# Table Layout

- TableLayout going to be arranged groups of views into **rows and columns**.
- Use the **<TableRow>** element to build a row in the table.
- Each row has **zero or more cells**; each cell can hold one View object
- It **don't display border lines** for their rows, columns, or cells.

<TableLayout>

Row 1		
Row 2 column 1	Row 2 column 2	Row 2 column 3
Row 3 column 1		Row 3 column 2

</ TableLayout>



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:collapseColumns	This specifies the zero-based index of the columns to collapse.
android:shrinkColumns	The zero-based index of the columns to shrink.
android:stretchColumns	The zero-based index of the columns to stretch.



# Example

<TableLayout

```
xmlns:android="http://schemas.android.com/apk/res/android" android:layout_width="fill_parent" android:layout_height="fill_parent">
```

<TableRow

```
    android:layout_width="fill_parent" android:layout_height="fill_parent">
```

```
        <!-- More GUI components go here -->
```

</TableRow>

```
<!-- More Table rows go here -->
```

</TableLayout>



**SATHYABAAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output

Time 10:25 AM

First Name \_\_\_\_\_

Last Name \_\_\_\_\_

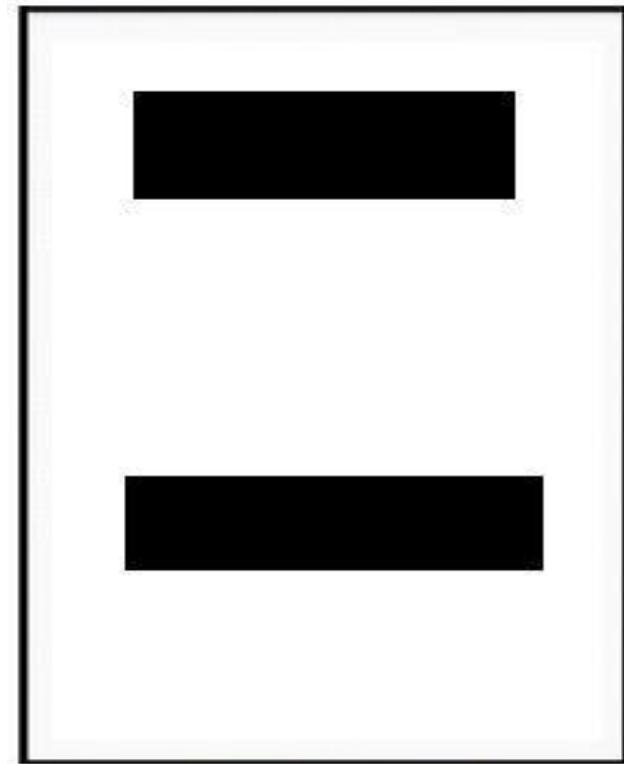
☆ ☆ ☆ ☆ ☆

**SUBMIT**



# Absolute Layout

- Absolute Layout lets you specify **exact locations** (x/y coordinates) of its children.
- Absolute layouts are **less flexible and harder to maintain** than other types of layouts without absolute positioning.



AbsoluteLayout



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:layout_x	This specifies the x-coordinate of the view.
android:layout_y	This specifies the y-coordinate of the view.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



<AbsoluteLayout

  xmlns:android=["http://schemas.android.com/apk/res/android"](http://schemas.android.com/apk/res/android)

    android:layout\_width="fill\_parent"

    android:layout\_height="fill\_parent">

      <Button android:layout\_width="100dp"

        android:layout\_height="wrap\_content"

        android:text="OK"

        android:layout\_x="50px"

        android:layout\_y="361px" />

<!-- More GUI components go here -->

</AbsoluteLayout>



# Output





# Frame Layout

- Frame Layout is designed to block out an area on the screen to **display a single item**.
- Generally, Frame Layout should be used to hold a **single child view**, because it can be difficult to organize child views in a way that's scalable to different screen sizes without the children overlapping each other.





# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:foreground	This defines the drawable to draw over the content and possible values may be a color value.
android:foregroundGravity	Defines the gravity to apply to the foreground drawable. The gravity defaults to fill. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.
android:measureAllChildren	Determines whether to measure all children or just those in the VISIBLE or INVISIBLE state when measuring. Defaults to false.



# Example

<FrameLayout

```
xmlns:android=http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="fill_parent"
```

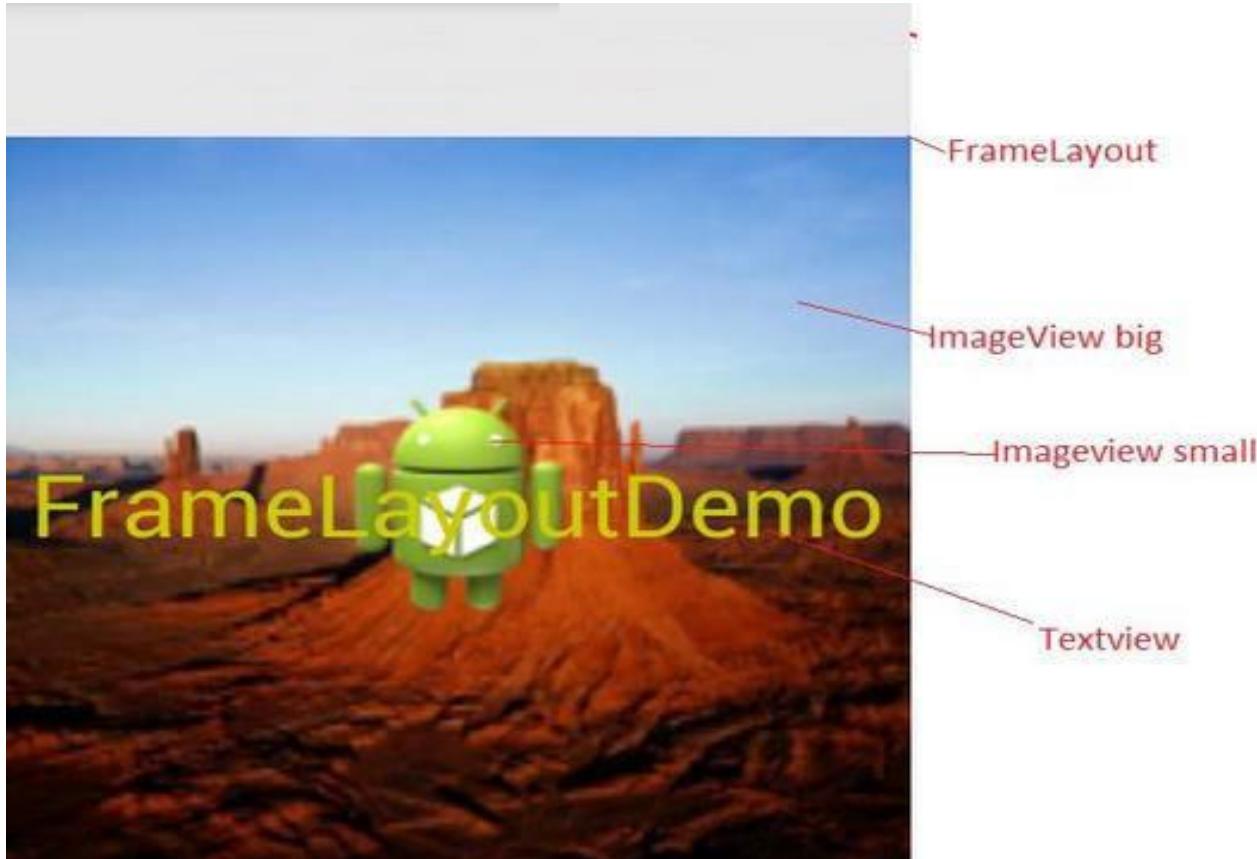
```
    android:layout_height="fill_parent">
```

```
        <!-- More GUI components go here -->
```

</FrameLayout>



# Output

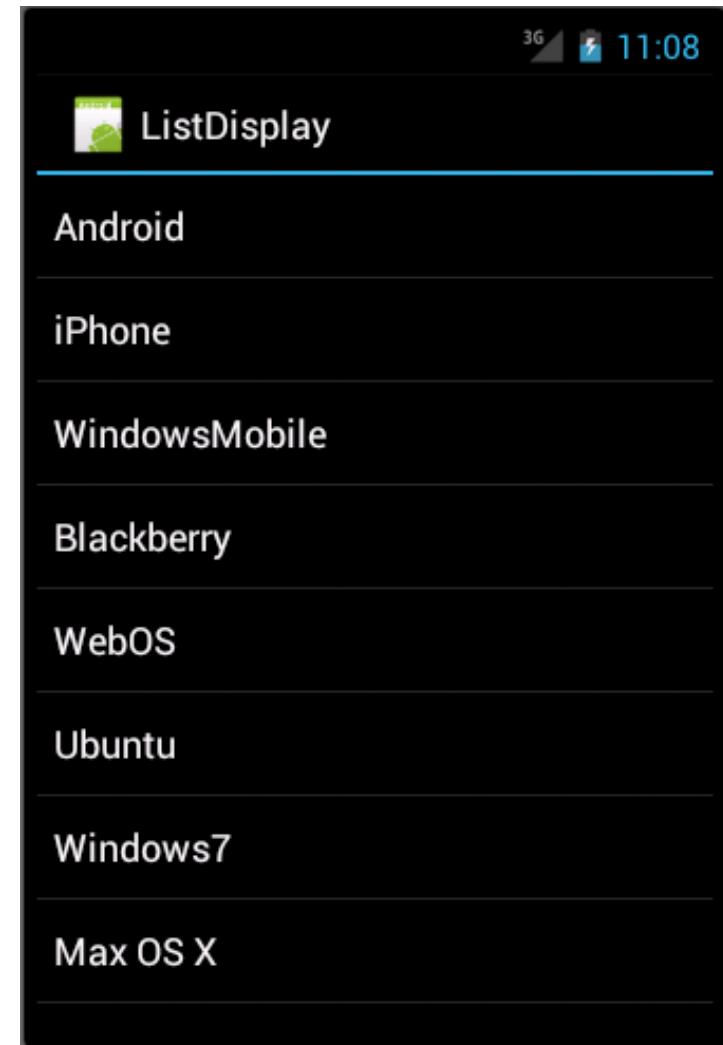


FRAME LAYOUT



# List View

- List View is a view which groups several items and display them in **vertical scrollable list**.
- The list items are automatically inserted to the list using an **Adapter** that pulls content from a source such as an array or database.





# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:divider	This is drawable or color to draw between list items.
android:dividerHeight	This specifies height of the divider. This could be in px, dp, sp, in, or mm.
android:entries	Specifies the reference to an array resource that will populate the ListView.
android:footerDividersEnabled	When set to false, the ListView will not draw the divider before each footer view. The default value is true.
android:headerDividersEnabled	When set to false, the ListView will not draw the divider after each header view. The default value is true.



# Example

<LinearLayout

```
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".ListActivity" >
```

```
    <ListView android:id="@+id/mobile_list"
              android:layout_width="match_parent"
              android:layout_height="wrap_content" >
        </ListView>
```

</LinearLayout>



# Output

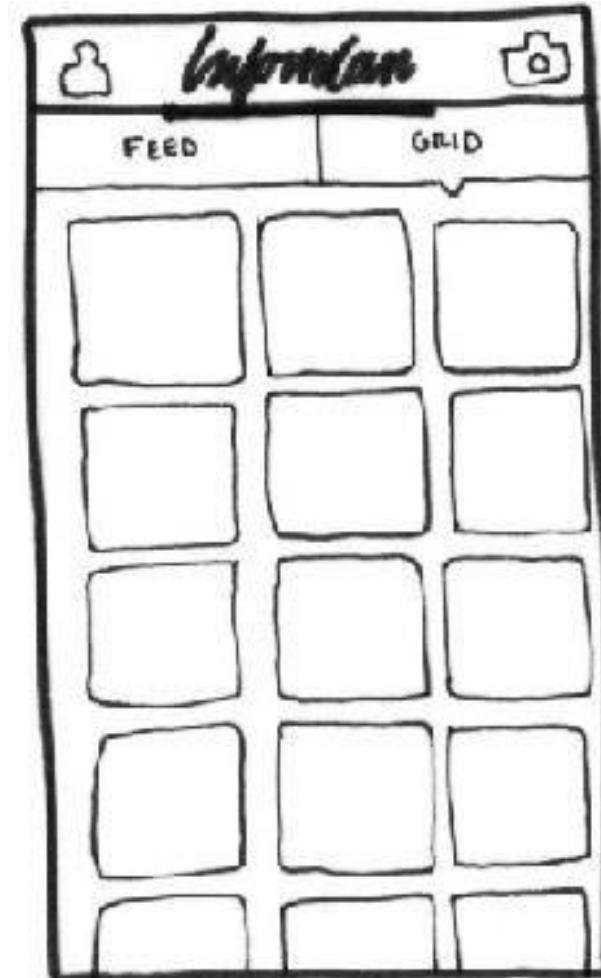


**LIST VIEW**



# Grid View

- Grid View shows items in **two-dimensional scrolling grid** (rows & columns)
- The grid items are not necessarily predetermined but they automatically inserted to the layout using a **ListAdapter**



GRID VIEW



## Grid View (con...)

- An adapter actually bridges between UI components and the data source that fill data into UI Component.
- Adapter can be used to supply the data to like spinner, list view, grid view etc.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Attribute	Description
android:id	This is the ID which uniquely identifies the layout.
android:columnWidth	This specifies the fixed width for each column. This could be in px, dp, sp, in, or mm.
android:gravity	Specifies the gravity within each cell. Possible values are top, bottom, left, right, center, center_vertical, center_horizontal etc.
android:horizontalSpacing	Defines the default horizontal spacing between columns. This could be in px, dp, sp, in, or mm.
android:numColumns	Defines how many columns to show.
android:verticalSpacing	Defines the default vertical spacing between rows. This could be in px, dp, sp, in, or mm.



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



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

```
<GridView
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```
        android:id="@+id/gridview"
```

```
        android:layout_width="fill_parent"
```

```
        android:layout_height="fill_parent"
```

```
        android:columnWidth="90dp"
```

```
        android:numColumns="auto_fit"
```

```
        android:verticalSpacing="10dp"
```

```
        android:horizontalSpacing="10dp"
```

```
        android:stretchMode="columnWidth"
```

```
        android:gravity="center" />
```



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output

---





## 7. UI Controls / Widgets

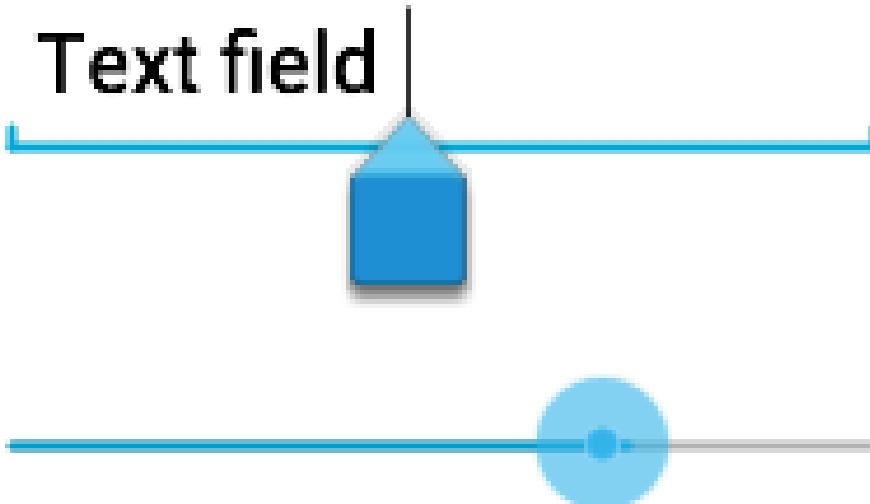
- Input controls are the **interactive components** in your app's user interface.
- Android provides a **wide variety of controls** you can use in your UI, such as buttons, text fields, seek bars, check box, zoom buttons, toggle buttons, and many more



Button



Text field



OFF

ON

**UI ELEMENTS**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC

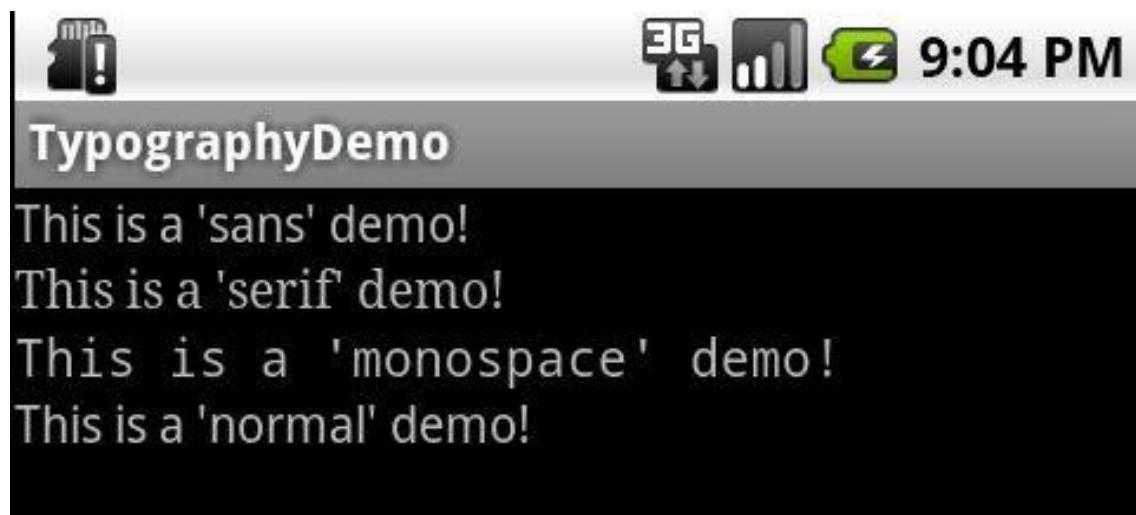


- TextView
- EditText
- Button
- ImageButton
- ToggleButton
- AutoCompleteTextVie w
- CheckBox
- RadioButton
- RadioGroup
- ProgressBar
- Spinner
- TimePicker
- DatePicker



# TextView Control

- A TextView **displays text to the user** and optionally allows them to edit it.
- A TextView is a **complete text editor**, however the basic class is configured to not allow editing.





Attribute	Description
android:id	This is the ID which uniquely identifies the control.
android:fontFamily	Font family (named by string) for the text.
android:inputType	The type of data being placed in a text field. Phone, Date, Time, Number, Password etc.
android:text	Text to display.
android:textAllCaps	Present the text in ALL CAPS. Possible value either "true" or "false".
android:textColor	Text color. May be a color value.
android:textSize	Size of the text. Recommended dimension type for text is "sp" for scaled-pixels.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



In XML:

<TextView

```
    android:id="@+id/text_id" android:layout_width="300dp"
    android:layout_height="200dp"
    android:capitalize="characters" android:text="hello_world"
    android:textColor="@android:color/holo_blue_dark"
    android:textColorHighlight="@android:color/primary_text_dark"
    android:layout_centerVertical="true"
    android:layout_alignParentEnd="true"
    android:textSize="50dp"/>
```

In JAVA:

```
TextView txtView = (TextView) findViewById(R.id.text_id);
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC

A screenshot of an Android application demonstrating different text styles. The app has a title bar with "demo" and three dots. The main screen shows the following text:

hello\_world

AndroidStyle

Hello World, AndroidStyleActivity!

mystyle

mystyle.20dp

**mystyle.20dp.bold**

**mystyle.30dp**

Android build-in style: TextAppearance

Android build-in style: TextAppearance.Inverse

Android build-in style: TextAppearance.Small

Android build-in style: TextAppearance.Medium

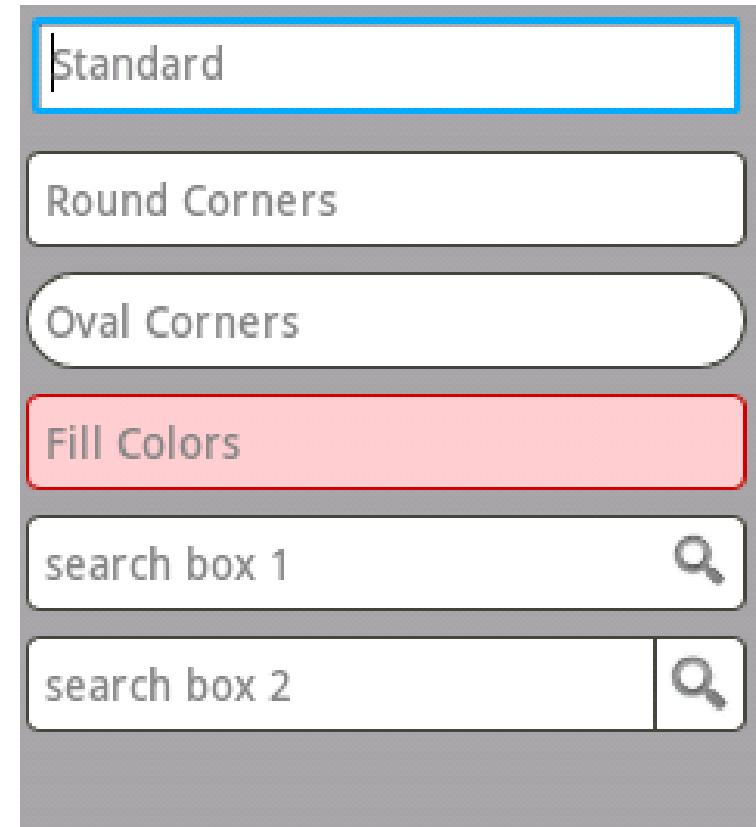
Android build-in style: TextAppearance.Large

The text "mystyle.20dp.bold" and "mystyle.30dp" are displayed in bold, while the other text is in regular weight. The background of the app's main screen is black, and the text is white or light gray.



# EditText Control

- A EditText is an overlay over TextView that configures itself to be **editable**.
- It is the predefined subclass of TextView that includes rich editing capabilities.



**STYLES OF EDIT TEXT**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Attribute	Description
android:autoText	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
android:drawableBottom	This is the drawable to be drawn below the text.
android:drawableRight	This is the drawable to be drawn to the right of the text.
android:editable	If set, specifies that this TextView has an input method.
android:text	This is the Text to display.



In XML:

<EditText

```
    android:id="@+id/edittext"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/button"
    android:layout_below="@+id/textView1"
    android:layout_marginTop="61dp" android:ems="10"
    android:text="@string/enter_text"
    android:inputType="text" />
```

In JAVA:

```
EditText eText = (EditText) findViewById(R.id.edittext);
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



demo

EditText

enter text

SHOW THE TEXT

# Output



## Button Control

- A Button is a Push-button which can be pressed, or clicked, by **the user** to perform an action.





# Attributes

Attribute	Description
android:autoText	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
android:drawableBottom	This is the drawable to be drawn below the text.
android:drawableRight	This is the drawable to be drawn to the right of the text.
android:editable	If set, specifies that this TextView has an input method.
android:text	This is the Text to display.



In XML:

# Example

```
<Button android:layout_width="wrap_content"  
       android:layout_height="wrap_content"  
       android:text="Button" android:id="@+id/button"  
       android:layout_alignTop="@+id/editText"  
       android:layout_alignLeft="@+id/textView1"  
       android:layout_alignStart="@+id/textView1"  
       android:layout_alignRight="@+id/editText"  
       android:layout_alignEnd="@+id/editText" />
```



In JAVA:

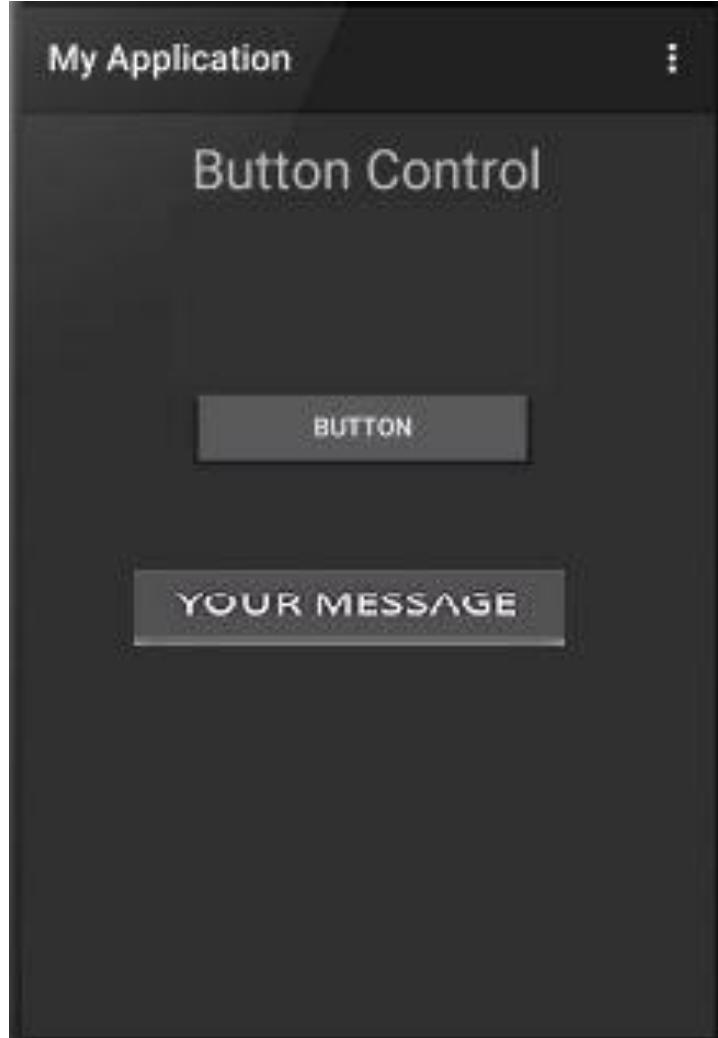
```
Button b1=(Button)findViewById(R.id.button);
b1.setOnClickListener(new View.OnClickListener()
{
    @Override
    public void onClick(View v)
    {
        Toast.makeText(MainActivity.this,"YOUR
MESSAGE",Toast.LENGTH_LONG).show();
    }
});
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output



# ImageButton Control

- A ImageButton is a AbsoluteLayout which enables you **to specify the exact location** of its children.
- This shows a **button with an image** (instead of text) that can be pressed or clicked by the user.



# Attributes

Attribute	Description
android:adjustViewBounds	Set this to true if you want the ImageView to adjust its bounds to preserve the aspect ratio of its drawable.
android:baseline	This is the offset of the baseline within this view.
android:baselineAlignBottom	If true, the image view will be baseline aligned with based on its bottom edge.
android:cropToPadding	If true, the image will be cropped to fit within its padding.
android:src	This sets a drawable as the content of this ImageView.



In XML:

# Example

```
<ImageButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/imageButton"  
    android:layout_centerVertical="true"  
    android:layout_centerHorizontal="true"  
    android:src="@drawable/abc"/>
```



In JAVA:

## Example (con...)

```
ImageButton imgButton =(ImageButton)
    findViewById(R.id.imageButton);
imgButton.setOnClickListener(new
View.OnClickListener()
{
    @Override public void onClick(View v)
    {
        Toast.makeText(getApplicationContext(),"Test
Image Button",Toast.LENGTH_LONG).show();
    }
});
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output



# ToggleButton Control

- A ToggleButton displays **checked/unchecked states** as a button.
- It is basically an **on/off button** with a light indicator.



**TOGGLE BUTTON**



# Attributes

Attribute	Description
android:disabledAlpha	This is the alpha to apply to the indicator when disabled.
android:textOff	This is the text for the button when it is not checked.
android:textOn	This is the text for the button when it is checked.



In XML:

```
<ToggleButton
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="On"  
    android:id="@+id/toggleButton1"  
    android:checked="true" />
```

```
<ToggleButton
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Off"  
    android:id="@+id/toggleButton2"  
    android:checked="true" />
```

## Example

```
<Button
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/button2"  
    android:text="ClickMe" />
```



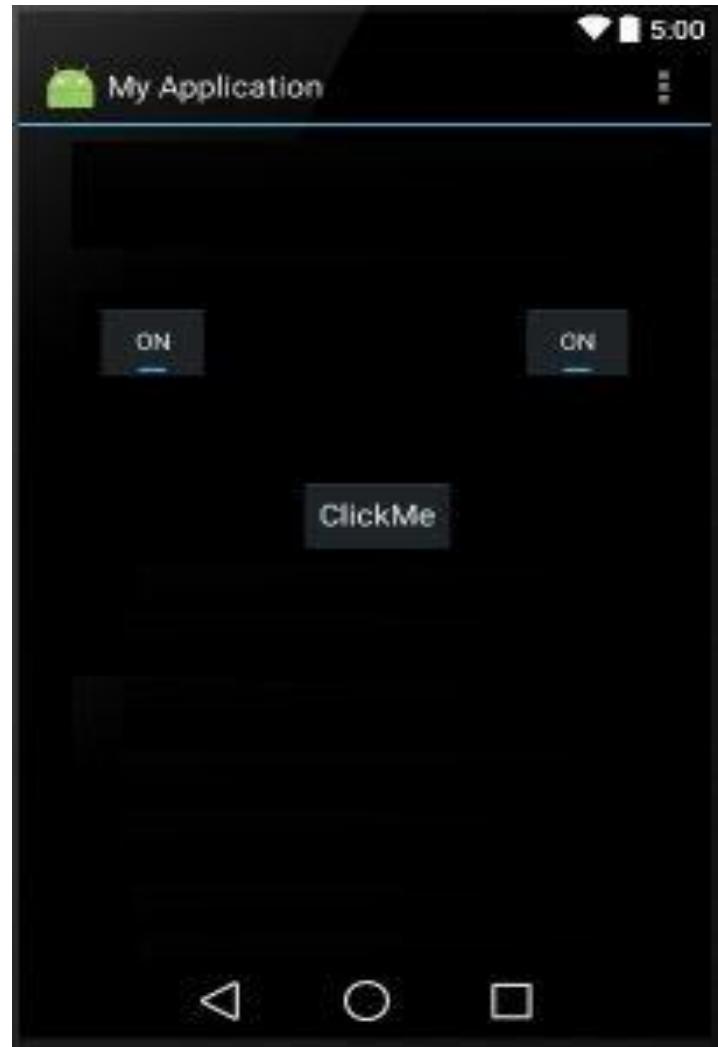
In JAVA:

```
ToggleButton tg1,tg2;  
Button b1;  
tg1=(ToggleButton)findViewById(R.id.toggleButton1);  
tg2=(ToggleButton)findViewById(R.id.toggleButton2);  
b1=(Button)findViewById(R.id.button2);  
b1.setOnClickListener(new View.OnClickListener() {  
    @Override public void onClick(View v) {  
        StringBuffer result = new StringBuffer();  
        result.append("You have clicked first ON Button").append(tg1.getText());  
        result.append("\nYou have clicked Second ON Button  
        ").append(tg2.getText());  
        Toast.makeText(MainActivity.this,result.toString(),Toast.LENGTH_SHORT)  
        .show(); } });
```

## Example (con...)



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output



# AutoCompleteTextView Control

- A AutoCompleteTextView is a view that is similar to EditText, except that it shows a **list of completion suggestions automatically** while the user is typing.
- The list of suggestions is displayed in **drop down menu**.
- The **user can choose** an item from there to replace the content of edit box with.



# Attributes

Attribute	Description
android:completionHintView	This defines the hint view displayed in the drop down menu.
android:completionThreshold	This defines the number of characters that the user must type before completion suggestions are displayed in a drop down menu.
android:dropDownAnchor	This is the View to anchor the auto-complete dropdown to.
android:dropDownHeight	This specifies the basic height of the dropdown.
android:dropDownSelector	This is the selector in a drop down list.
android:dropDownWidth	This specifies the basic width of the dropdown.
android:popupBackground	This sets the background.



In XML:

# Example

```
<AutoCompleteTextView  
    android:id="@+id/autoCompleteTextView1  
    " android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_alignLeft="@+id/textView2"  
    android:layout_below="@+id/textView2"  
    android:layout_marginTop="54dp"  
    android:ems="10" />
```



In JAVA:

```
AutoCompleteTextView autocomplete; String[]  
arr = { "Paris,France", "PA,United  
States", "Parana,Brazil", "Padua,Italy",  
"Pasadena,CA,United States";  
  
autocomplete = (AutoCompleteTextView)  
        findViewById(R.id.autoCompleteTextView1);  
  
ArrayAdapter<String> adapter = new  
        ArrayAdapter<String>  
        (this, android.R.layout.select_dialog_item, arr);  
autocomplete.setThreshold(2);  
autocomplete.setAdapter(adapter);
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC

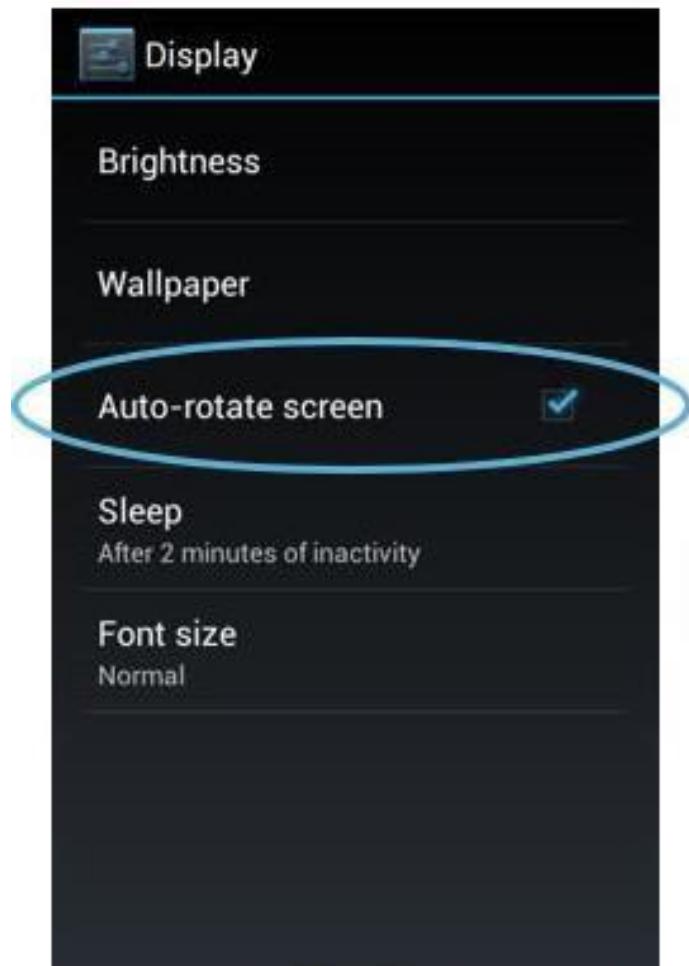


# Output



# CheckBox Control

- A CheckBox is an **on/off switch** that can be toggled by the user.
- To use check-boxes when presenting users with a **group of selectable options** that are not mutually exclusive.





# Attributes

Attribute	Description
android:autoText	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.
android:drawableBottom	This is the drawable to be drawn below the text.
android:drawableRight	This is the drawable to be drawn to the right of the text.
android:editable	If set, specifies that this TextView has an input method.
android:text	This is the Text to display.



In XML:

# Example

```
<CheckBox android:id="@+id/checkBox1"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Do you like android"
    android:checked="false" />

<CheckBox android:id="@+id/checkBox2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Do you like android "
    android:checked="false" />
```



In JAVA:

## Example (con...)

```
CheckBox ch1,ch2; Button b1,b2;
```

```
ch1=(CheckBox)findViewById(R.id.checkBox1);
```

```
ch2=(CheckBox)findViewById(R.id.checkBox2);
```

```
b1=(Button)findViewById(R.id.button);
```

```
b1.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
public void onClick(View v) {
```

```
    StringBuffer result = new StringBuffer();
```

```
    result.append("Thanks : ").append(ch1.isChecked());
```

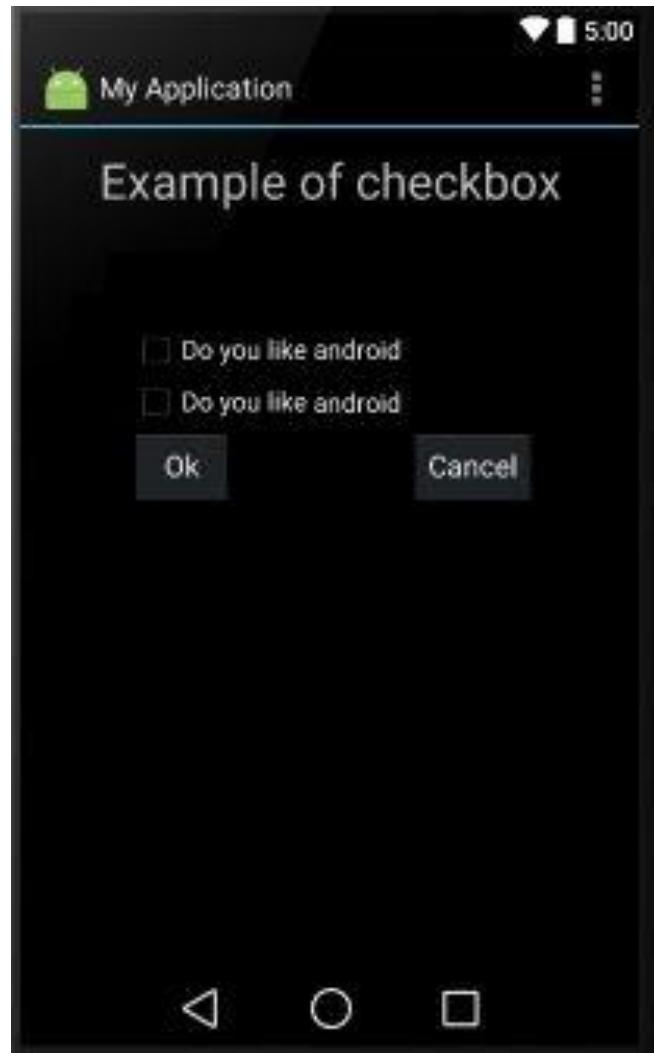
```
    result.append("\nThanks: ").append(ch2.isChecked());
```

```
    Toast.makeText(MainActivity.this, result.toString(),
```

```
    Toast.LENGTH_LONG).show(); }
```



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output



# RadioButton Control

- A RadioButton has **two states**: either checked or unchecked.
- This allows the user to **select one option** from a set.

ATTENDING?

---

Yes       Maybe       No

RADIO BUTTON



In XML:

<RadioGroup

<RadioButton

    android:text="JAVA"

<RadioButton

    android:text="HTML"

    android:id="@+id radioButton1"

    android:id="@+id/radioButton"

3"

    android:checked="false" />

    android:id="@+id radioButton2"

    </RadioGroup>

    android:checked="false" />

<RadioButton

    android:text="ANDROID"

    android:id="@+id radioButton3"

    android:checked="false" />



In JAVA:

## Example (con...)

RadioButton rb1; RadioGroup rg1; Button b1;

addListenerRadioButton();

private void addListenerRadioButton() {

rg1 = (RadioGroup) findViewById(R.id.radioGroup);

b1 = (Button) findViewById(R.id.button1);

b1.setOnClickListener(new View.OnClickListener() {

@Override public void onClick(View v) {

int selected=rg1.getCheckedRadioButtonId();

rb1=(RadioButton)findViewById(selected);

Toast.makeText(MainActivity.this,rb1.getText(),Toast.LENGTH\_LONG).show(); } }); }



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# Output



# RadioGroup Control

- A RadioGroup class is used for **set of radio buttons**.
- If we **check one radio button** that belongs to a radio group, it automatically **unchecks** any previously checked radio button within the same group.

(Refer RadioButton)



# Progress Bar Control

- Progress bars are used to **show progress of a task.**
- A class called **ProgressDialog** that allows you to create progress bar.
- **Syntax:**

```
ProgressDialog progress = new  
ProgressDialog(this);
```

- For **example**, when you are uploading or downloading something from the internet, it is better to show the progress of download/upload to the user.



# ProgressDialog class methods

Methods	Description
<b>getMax()</b>	This method returns the maximum value of the progress.
<b>incrementProgressBy(int diff)</b>	This method increments the progress bar by the difference of value passed as a parameter.
<b>setIndeterminate(boolean indeterminate)</b>	This method sets the progress indicator as determinate or indeterminate.
<b>setMax(int max)</b>	This method sets the maximum value of the progress dialog.
<b>setProgress(int value)</b>	This method is used to update the progress dialog with some specific value.
<b>show(Context context, CharSequence title, CharSequence message)</b>	This is a static method, used to display progress dialog.



In XML:

# Example

```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="DOWNLOAD"  
    android:onClick="download"  
    android:id="@+id/button1"  
    android:layout_marginLeft="125dp"  
    android:layout_marginStart="125dp"  
    android:layout_centerVertical="true" />
```



In JAVA:

## Example (con...)

Button b1;

```
private ProgressDialog progress;
```

```
b1 = (Button) findViewById(R.id.button1);
```

```
public void download(View view){
```

```
    progress=new ProgressDialog(this);
```

```
    progress.setMessage("Downloading Music");
```

```
    progress.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL); progress.setIndeterminate(true);
```

```
    progress.setProgress(0);
```

```
    progress.show();
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



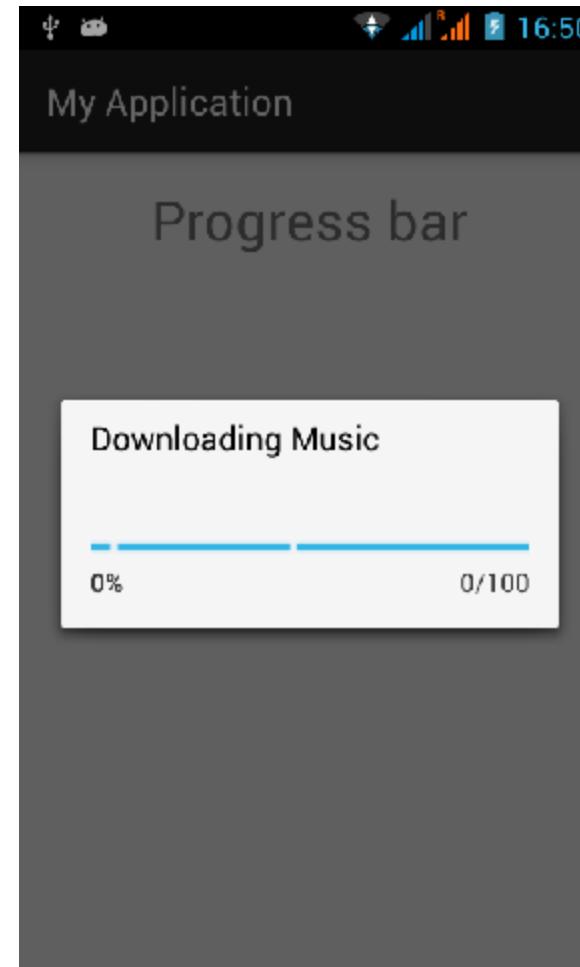
```
final int totalProgressTime = 100;  
final Thread t = new Thread() {  
    @Override public void run() {  
        int jumpTime = 0;  
        while(jumpTime < totalProgressTime) {  
            try {  
                sleep(200);  
                jumpTime += 5;  
                progress.setProgress(jumpTime); }  
            catch (InterruptedException e) {}  
        } } };  
t.start(); }
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





# Spinner Control

- Spinner allows you to select an item from a drop down menu.

Select a Month

January ▾



In XML:

# Example

```
<Spinner android:id="@+id/spinner"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:prompt="@string/spinner_title"/>
```

In JAVA:

```
Spinner spinner = (Spinner) findViewById(R.id.spinner);
spinner.setOnItemSelectedListener(this);

List<String> categories = new ArrayList<String>();
categories.add("Automobile");
categories.add("Business Services");
```



## Example (con...)

```
categories.add("Computers");
```

```
categories.add("Education");
```

```
categories.add("Personal");
```

```
categories.add("Travel");
```

```
ArrayAdapter<String> dataAdapter = new
    ArrayAdapter<String>(this,
        android.R.layout.simple_spinner_item, categories);
dataAdapter.setDropDownViewResource(android.R.layout.
    simple_spinner_dropdown_item);
spinner.setAdapter(dataAdapter);
```



## Example (con...)

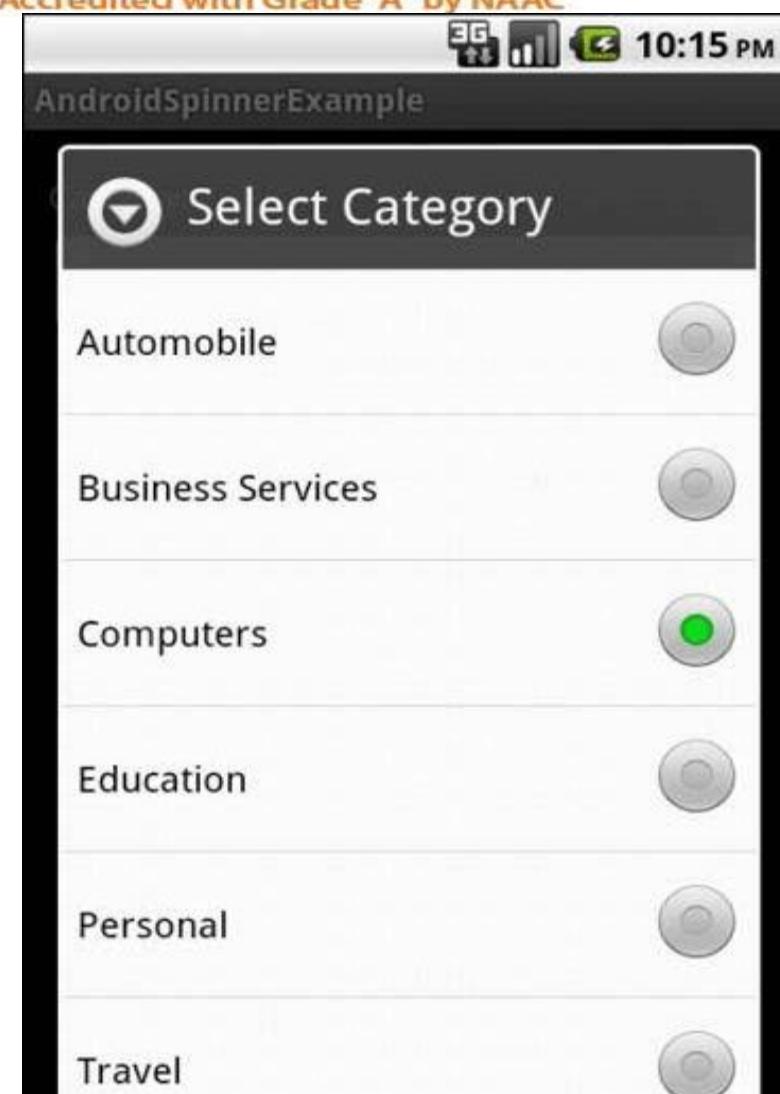
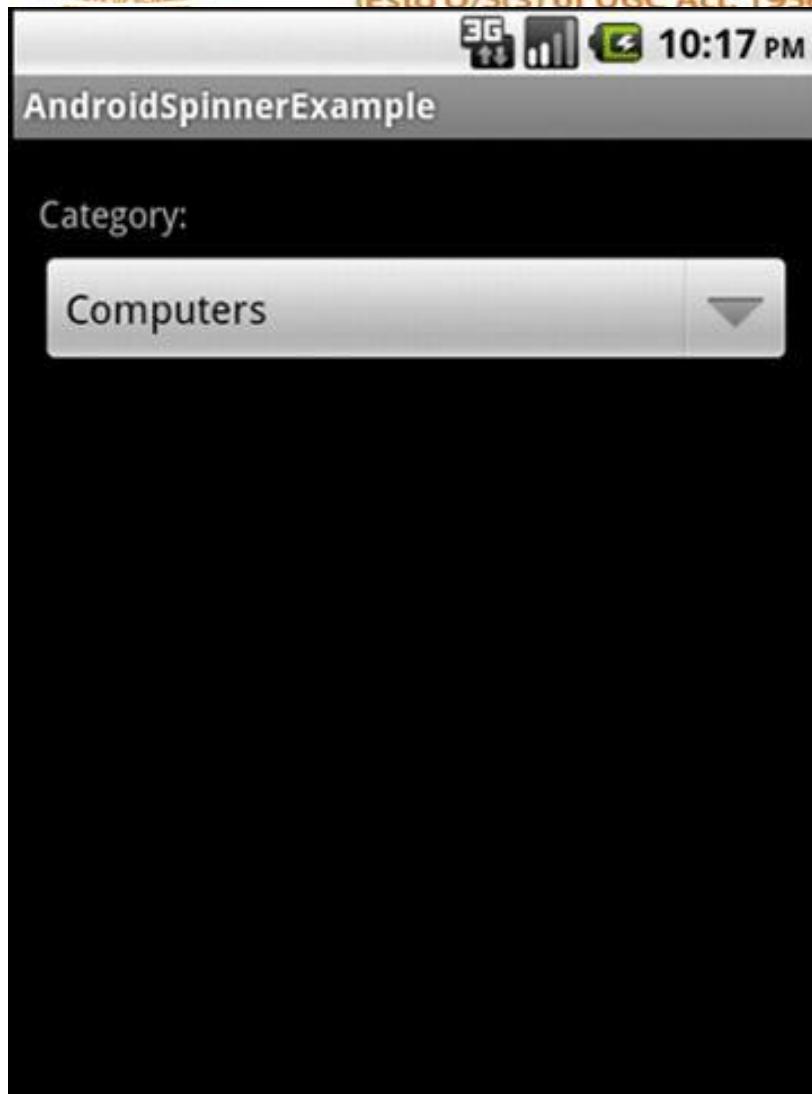
```
public void onItemSelected(AdapterView<?>
    parent, View view, int position, long id) {
String item =
    parent.getItemAtPosition(position).toString();
Toast.makeText(parent.getContext(), "Selected: "
    + item, Toast.LENGTH_LONG).show();
}
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

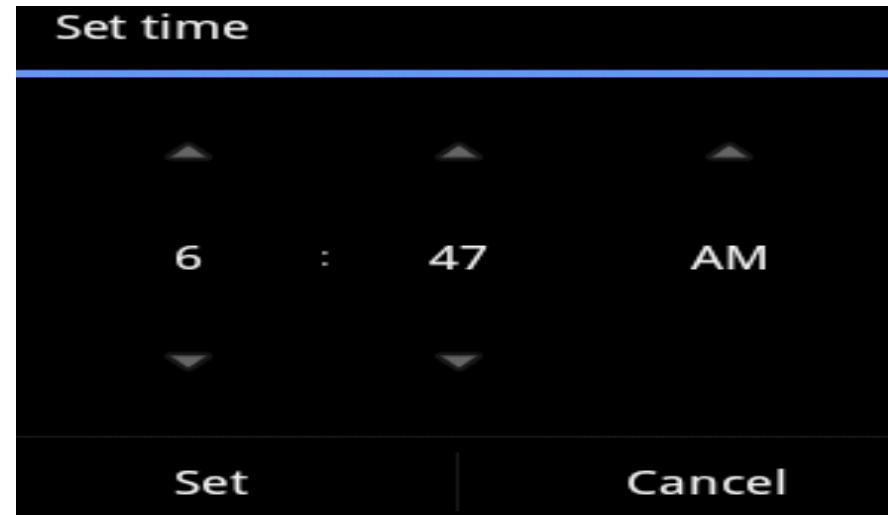
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





# TimePicker Control

- Time Picker allows you to select the **time of day** in either 24 hour or AM/PM mode.
- The time consists of **hours, minutes and clock format**.
- Android provides this functionality through **TimePicker class**.





# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Methods	Description
<b>is24HourView()</b>	This method returns true if this is in 24 hour view else false
<b>isEnabled()</b>	This method returns the enabled status for this view
<b>setCurrentHour(Integer currentHour)</b>	This method sets the current hour
<b>setCurrentMinute(Integer currentMinute)</b>	This method sets the current minute
<b>setEnabled(boolean enabled)</b>	This method set the enabled state of this view
<b>setIs24HourView(Boolean is24HourView)</b>	This method set whether in 24 hour or AM/PM mode
<b>setOnTimeChangedListener(TimePicker.OnTimeChangedListener onTimeChangedListener)</b>	This method Set the callback that indicates the time has been adjusted by the user



In XML:

## Example

```
<TimePicker android:id="@+id/timePicker1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

In JAVA:

```
TimePicker timePicker1;
```

```
timePicker1 = (TimePicker)findViewById(R.id.timePicker1);
```

```
int hour = timePicker1.getCurrentHour();
```

```
int min = timePicker1.getCurrentMinute();
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd US/2002) - AICTE Approved & Accredited with Grade "A" by NAAC

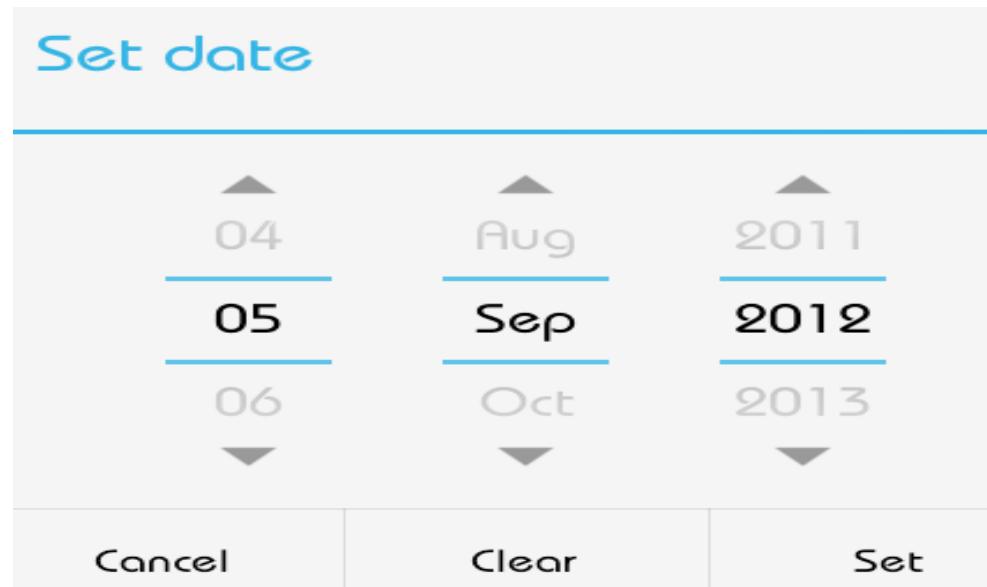


# Output



# DatePicker Control

- Date Picker allows you to **select the date** consisting of day, month and year in your custom user interface.
- Android provides **DatePicker** and **DatePickerDialog** components.





# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)



Methods	Description
<b>getDayOfMonth()</b>	This method gets the selected day of month
<b>getMonth()</b>	This method gets the selected month
<b>getYear()</b>	This method gets the selected year
<b>setMaxDate(long maxDate)</b>	This method sets the maximal date supported by this DatePicker in milliseconds
<b>setMinDate(long minDate)</b>	This method sets the minimal date supported by this NumberPicker in milliseconds
<b>setSpinnersShown(boolean shown)</b>	This method sets whether the spinners are shown
<b>updateDate(int year, int month, int dayOfMonth)</b>	This method updates the current date
<b>getCalendarView()</b>	This method returns calendar view
<b>getFirstDayOfWeek()</b>	This Method returns first day of the week



In JAVA:

# Example

```
DatePicker datePicker;  
Calendar calendar;  
int year, month, day;  
calendar = Calendar.getInstance();  
year = calendar.get(Calendar.YEAR);  
month = calendar.get(Calendar.MONTH);  
day = calendar.get(Calendar.DAY_OF_MONTH);
```



## 8. Event Handling

- Events are a **useful way to collect data about a user's interaction** with interactive components of Applications.
- Like button **presses or screen touch** etc.
- The Android framework maintains an event queue as **first-in, first-out (FIFO) basis**.
- Capture these events in program and **take appropriate action** as per requirements.



# Event Handling (con...)

- Event Management

- Event Listeners

- An event listener is an interface in the View class that contains a **single callback method**.
    - These methods will be called by the Android framework when the View to which the listener has been registered is **triggered by user interaction** with the item in the UI.



# Event Handling (con...)

## – Event Handlers

- When an event happens and we have registered in the event listener for the event, the event listener calls the Event Handlers, which is the method that actually handles the event.

## – Event Listeners Registration

- Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event



## Event Listeners & Event Handlers

Event Handler	Event Listener	Description
onClick()	OnClickListener()	This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc.
onLongClick()	OnLongClickListener()	This is called when the user either clicks or touches or focuses upon any widget like button, text, image etc. for one or more seconds.
onFocusChange()	OnFocusChangeListener()	This is called when the widget loses its focus.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



onKey()	<b>OnFocusChangeListener()</b>	This is called when the user is focused on the item and presses or releases a hardware key on the device.
onTouch()	<b>OnTouchListener()</b>	This is called when the user presses the key, releases the key, or any movement gesture on the screen.
onMenuItemClick()	<b>OnMenuItemClickListener()</b>	This is called when the user selects a menu item.
onCreateContextMenu()	<b>onCreateContextMenuListener()</b>	This is called when the context menu is being built(as the result of a sustained "long click").



# Event Listeners Registration

- Event Registration is the process by which an **Event Handler** gets registered with an **Event Listener** so that the handler is called when the Event Listener fires the event.
- **Top 3 ways** are,
  - Using an **Anonymous Inner Class**
  - Activity class **implements the Listener interface.**
  - Using **Layout file** activity\_main.xml to specify event handler directly.



# Example

- Using an **Anonymous Inner Class**

Button b1;

```
b1=(Button)findViewById(R.id.button);
```

```
b1.setOnClickListener(new View.OnClickListener())
```

```
{ @Override
```

```
public void onClick(View v) {
```

```
    TextView txtView = (TextView)
```

```
    findViewById(R.id.textView);
```

```
    txtView.setTextSize(25); } );
```



## Example (con...)

- Activity class implements the Listener interface

```
BtnListener listener = new BtnListener();
```

```
((Button)
```

```
findViewById(R.id.btnNum0Id)).setOnClickListener(listener);
```

```
private class BtnListener implements
```

```
OnClickListener { // On-click event handler for all  
the buttons @Override public void onClick(View  
view) {
```

```
    //ToDo the code here....
```

```
}
```



## Example (con...)

- Using **Layout file** activity\_main.xml to specify event handler directly
- In XML

<Button

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Small font"
    android:id="@+id/button"
    android:onClick="Font_Change"/>
```



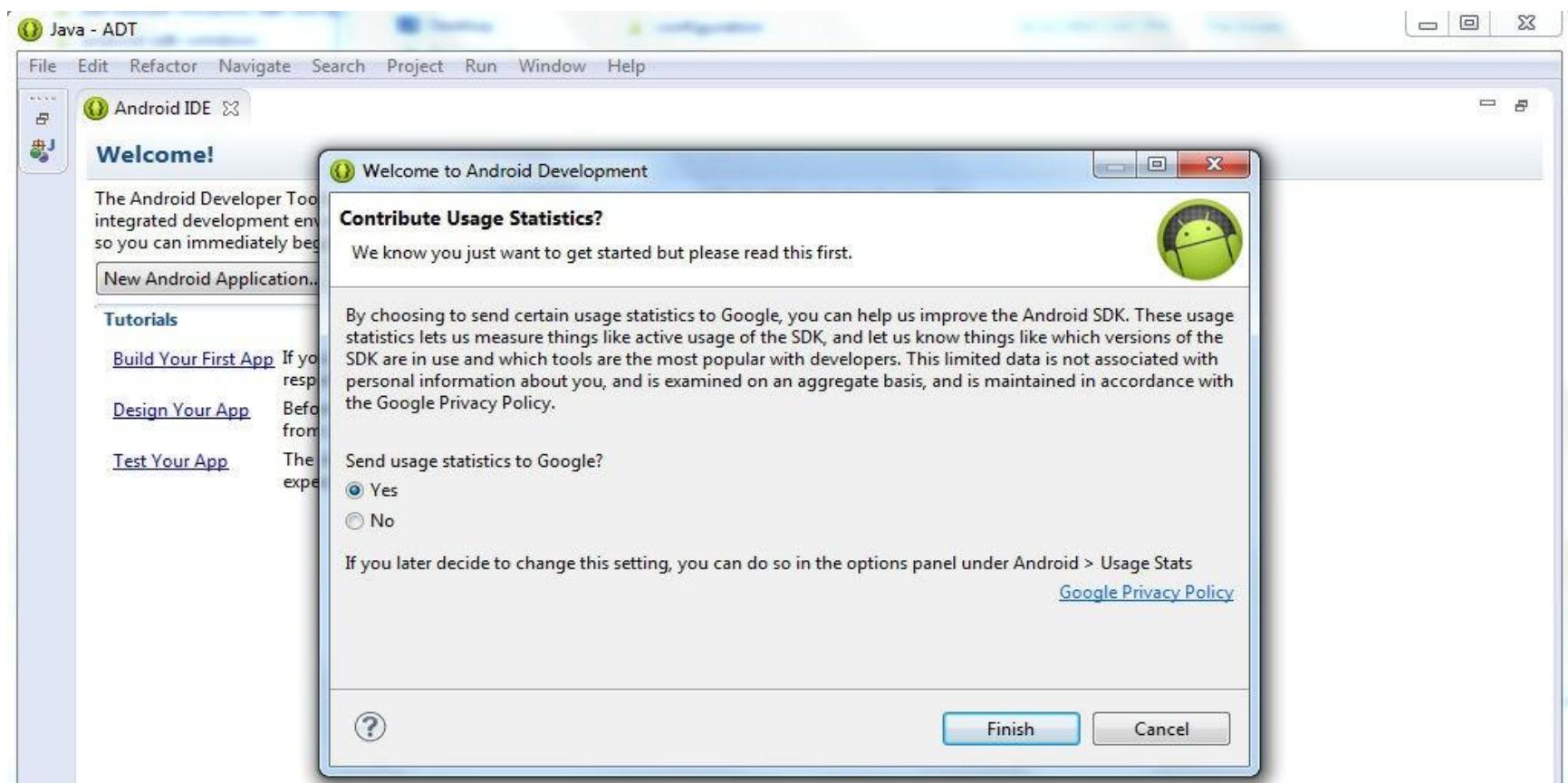
# Example (con...)

- In JAVA

```
public void Font_Change(View v) {  
    TextView txtView = (TextView)  
        findViewById(R.id.textView);  
    txtView.setTextSize(25);  
}
```



# 9. Tools - Eclipse IDE





# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**



Java - ADT

File Edit Refactor Navigate Search Project Run Window Help

Quick Access Java

Package Explorer    An outline is not available.

Problems @ Javadoc Declaration

0 items

Description	Resource	Path	Local
-------------	----------	------	-------



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Java - ADT

File Edit Refactor Navigate S...

Package Ex... X □ □

Java (default) Java Browsing Java Type Hierarchy Pixel Perfect Resource Team Synchronizing Tracer for OpenGL ES X XML

An outline is not available.

OK Cancel

Description Resource Path Local

This screenshot shows the Eclipse IDE interface for Java Development (ADT). A central dialog box titled "Open Perspective" is displayed, listing various perspectives: C/C++, DDMS, Debug, Git Repository Exploring, Hierarchy View, Java (default), Java Browsing, Java Type Hierarchy, Pixel Perfect, Resource, Team Synchronizing, Tracer for OpenGL ES, and XML. The "Java (default)" perspective is currently selected. In the background, the Java perspective is active, showing a message "An outline is not available." and a table with columns for Description, Resource, Path, and Local.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**



Java - ADT

File Edit Refactor Source Navigate Search Project Run Window Help

New Alt+Shift+N ▾

- Java Project
- Android Application Project
- Project...
- Package
- Class
- Interface
- Enum
- Annotation
- Source Folder
- Java Working Set
- Folder
- File
- Untitled Text File
- Android XML File
- JUnit Test Case
- Example...
- Other... Ctrl+N

Open File...

Close Ctrl+W

Close All Ctrl+Shift+W

Save Ctrl+S

Save As...

Save All Ctrl+Shift+S

Revert

Move...

Rename... F2

Refresh F5

Convert Line Delimiters To ▾

Print... Ctrl+P

Switch Workspace ▾

Restart

Import...

Export...

Properties Alt+Enter

Exit

Java DDMS



New Android Application

## New Android Application

The prefix 'com.example.' is meant as a placeholder and should not be used



Application Name:

Project Name:

Package Name:

Minimum Required SDK:

Target SDK:

Compile With:

Theme:

Choose the lowest version of Android that your application will support. Lower API levels target more devices, but means fewer features are available. By targeting API 8 and later, you reach approximately 95% of the market.



< Back

Next >

Finish

Cancel



New Android Application



## New Android Application

Configure Project

Create custom launcher icon

Create activity

Mark this project as a library

Create Project in Workspace

Location: D:\Users\veera\workspace\MyFirstApp

Browse...

### Working sets

Add project to working sets

Working sets:



Select...



< Back

Next >

Finish

Cancel



New Android Application



## Configure Launcher Icon

Configure the attributes of the icon set

Foreground:

Image File: launcher\_icon

Trim Surrounding Blank Space

Additional Padding:



0%

Foreground Scaling:

Shape

Background Color:





New Android Application



## Configure Launcher Icon

Configure the attributes of the icon set

Foreground: **Image** Clipart Text

Image File: C:\Users\Public\Pictures\Sample Pictures\ [Browse...](#)

Trim Surrounding Blank Space

Additional Padding:



%

Foreground Scaling: **Crop** Center

Shape **None** Square Circle

Background Color:

Preview:

mdpi:



hdpi:



xhdpi:



xxhdpi:





### Create Activity

Select whether to create an activity, and if so, what kind of activity.

Create Activity

**Blank Activity**

Fullscreen Activity

Master/Detail Flow



#### Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.



< Back

Next >

Finish

Cancel



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



## Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.



Activity Name

Layout Name

Navigation Type

The name of the activity class to create



< Back

Next >

Finish

Cancel



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Java - MyFirstApp/src/com/example/myfirstapp/MainActivity.java - ADT

File Edit Refactor Source Navigate Project Run Window Help

Quick Access Java DDMS

Package Explorer    MainActivity.java    Outline    Problems

MyFirstApp

- src
- com.example
- gen [Generated]
- com.example
- Android 4.4
- Android Private L
- assets
- bin
- res
- AndroidManifest.xml
- libs
- res
- drawable-hdpi
- drawable-ldpi
- drawable-mdpi
- drawable-xhdpi

```
activity_main.xml MainActivity.java R.java
```

```
package com.example.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @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;
    }
}
```

com.example.MainActivity

- onCreate
- onCreateOptionsMenu

Problems @ Javadoc Declaration

0 items

Description	Resource	Path	Local



# SATHYABAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)



Java - MyFirstApp/src/com/example/myfirstapp/MainActivity.java - ADT

File Edit Refactor Source Navigate Project Run Window Help

New Window  
New Editor  
Hide Toolbar  
Open Perspective  
Show View  
Customize Perspective...  
Save Perspective As...  
Reset Perspective...  
Close Perspective  
Close All Perspectives  
Navigation  
Android SDK Manager  
Android Virtual Device Manager  
 Run Android Lint  
Preferences

Package Explorer    activity\_main.xml    Java    DDMS

MyFirstApp

src

com.example

MainActivity

Generated J

com.example

BuildCont

R.java

Android 4.4

Android Private L

assets

bin

res

AndroidMani

libs

res

drawable-hd

drawable-ldp

drawable-md

drawable-xhc

activity\_main.xml

MainActivity

package com.example;

import android.os.B

public class MainAc

@Override  
protected void  
super.onCreate()  
setContentView()

@Override  
public boolean

Problems    @ Javadoc

0 items

Description	Resource	Path	Locat
-------------	----------	------	-------



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)



File Edit Refactor Source

Android Virtual Device Manager

Android Virtual Devices

List of existing Android Virtual Devices

AVD Name	Type
--	

AVD Name: Testing

Device: 3.2" QVGA (ADP2) (320 × 480: mdpi)

Target: Android 4.4 - API Level 19

CPU/ABI: ARM (armeabi-v7a)

Keyboard:  Hardware keyboard present

Skin:  Display a skin with hardware controls

Front Camera: None

Back Camera: None

Memory Options:

RAM: 512      VM Heap: 16

Internal Storage: 200 MiB

SD Card:

Size: 40 MiB  
 File:

Emulation Options:

Snapshot       Use Host GPU

Refresh



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Android Virtual Device Manager

Android Virtual Devices    Device Definitions

List of existing Android Virtual Devices located at C:\Users\Meabi-v...:

AVD Name	Target Name
Testing	Android 4.4

**Launch Options**

Skin: 320x480  
Density: Medium (160)  
 Scale display to real size

Screen Size (in): 3.2  
Monitor dpi: 96  
Scale: default

Wipe user data  
 Launch from snapshot  
 Save to snapshot

**Buttons:** Launch    Cancel    Refresh

A valid Android Virtual Device.  
 A repairable Android Virtual Device.  
 An Android Virtual Device that failed to load. Click 'Details' to see the error.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Android Virtual Device Manager

Android Virtual Devices    Device Definitions

List of existing Android Virtual Devices located at C:\Users\veera\.android\avd

AVD Name
Testing

New...    Edit...    Delete...    Repair...    Details...    Start...    Refresh

**Starting Android Emulator**

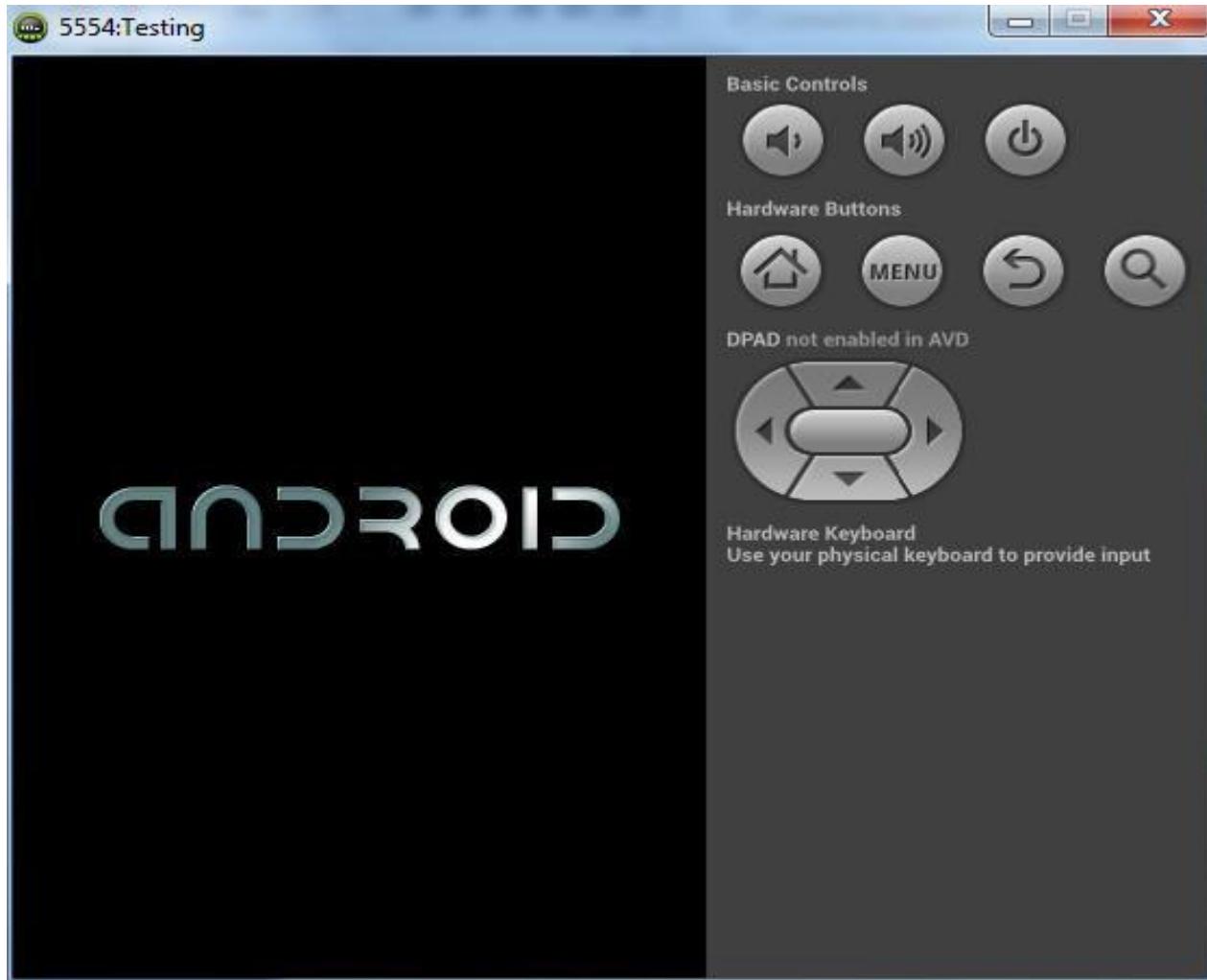
Starting emulator for AVD 'Testing'

Starting emulator for AVD 'Testing'

✓ A valid Android Virtual Device.    ⚡ A repairable Android Virtual Device.  
✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



# AVD – Emulator



# Configure the Logcat

**Auto Monitor Logcat**

Would you like ADT to automatically monitor logcat output for messages from applications in the workspace?

No, do not monitor logcat output.

Yes, monitor logcat and display logcat view if there are messages with priority higher than:

**OK**



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



Java - MyFirstApp/src/com/example/myfirstapp/MainActivity.java - ADT

File Edit Source Refactor Navigate Search Project Run Window Help

Quick Access

Java DDMS

Package Explorer

- MyFirstApp
  - src
    - com.example
    - MainActivity
  - gen [Generated J]
  - com.example
    - BuildConfig
    - R.java
  - Android 4.4
  - Android Private L
  - assets
  - bin
    - dexedLibs
    - res
    - AndroidManifest.xml
    - classes.dex
    - MyFirstApp.apk
    - resources.apk
  - libs
  - res

activity\_main.xml MainActivity.java

```
package com.example.myfirstapp;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @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;
    }
}
```

O... Java DDMS

com.example.MainActivity

- onCreate
- onCreateOptionsMenu

Problems Declaration LogCat

Saved Filters All messages com.example

Search for messages. Accepts Java regexes. Prefix: verbose

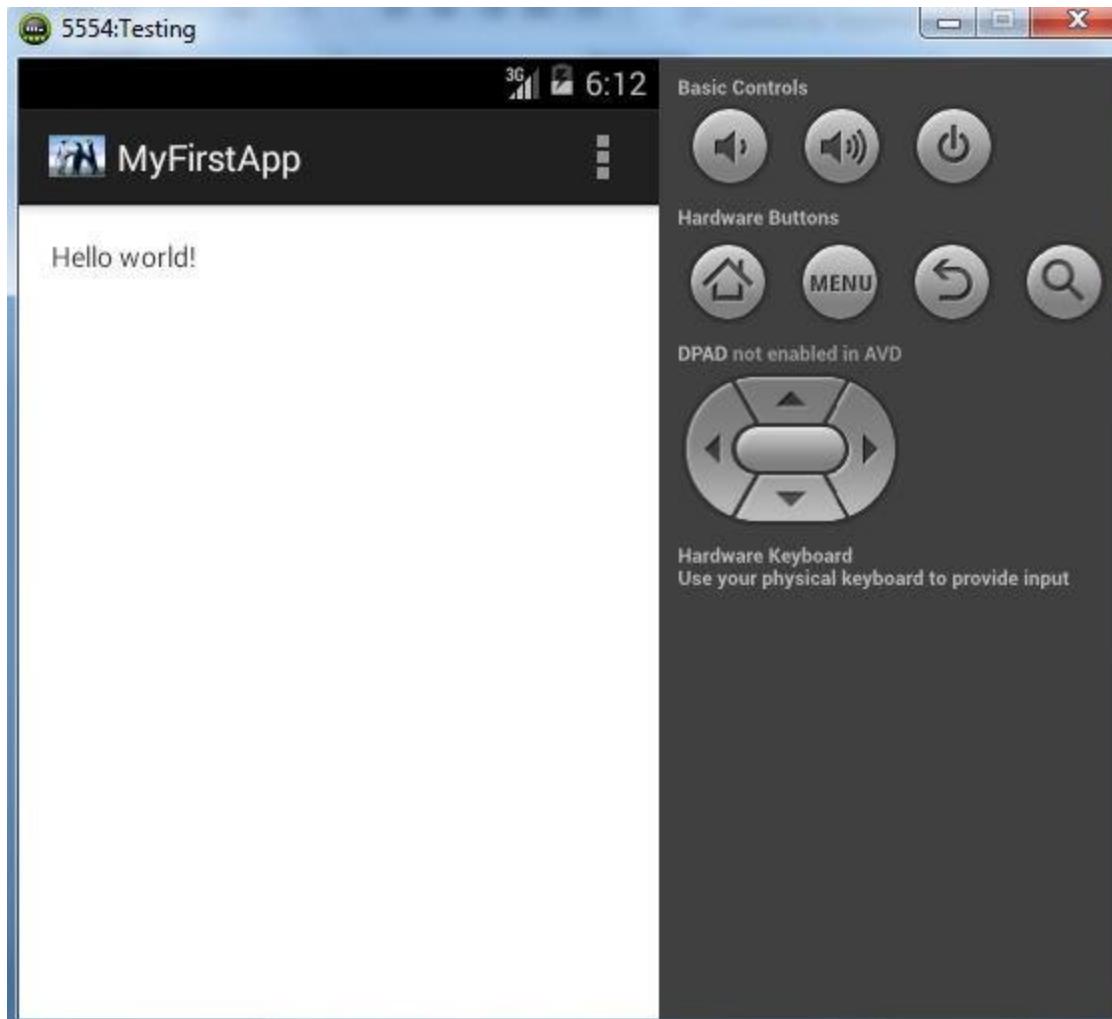
L...	Time	PID	TID	Application	Tags
D	06-22 06:10:4...	1134	1134	com.example.myf...	gr...



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC

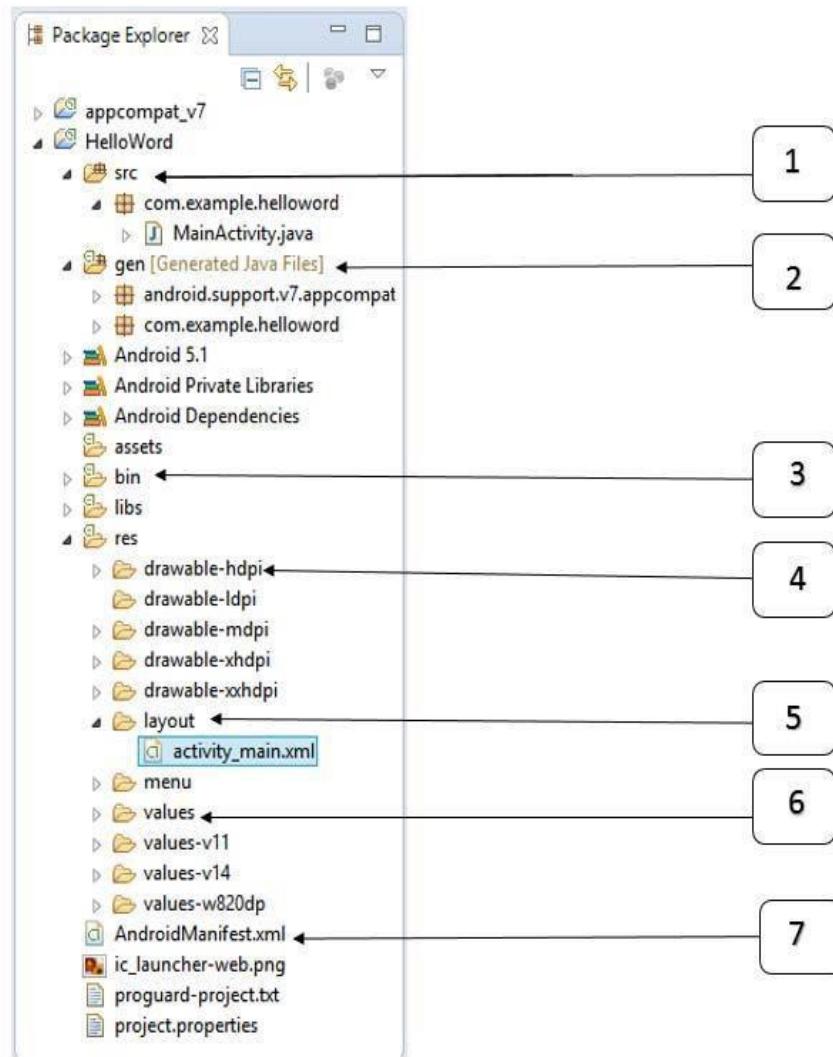




# SATHYABAAMA

## INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC





# Application Structure (con...)

1. src
2. gen
3. bin
4. res/drawable-hdpi
5. res/layout
6. res/values
7. AndroidManifest.xml



# Application Structure (con...)

## 1. src

- This contains the **.java source files** for your project.
- By default, it includes an ***MainActivity.java*** source file having an activity class that runs when your app is launched using the app icon.

## 2. gen

- This contains the **.R file**, a compiler-generated file that references all the resources found in your project.
- User should not modify this file.



# Application Structure (con...)

- bin
  - This folder contains the **Android package files .apk** built by the ADT during the build process and everything else needed to run an Android application.
- res/drawable-hdpi
  - This is a directory for **drawable objects** that are designed for high-density screens.
- res/layout
  - This is a directory for files that define your **app's user interface**.



# Application Structure (con...)

- res/values
  - This is a directory for other various **XML files** that contain a collection of resources, such as **strings and colours** definitions.
- AndroidManifest.xml
  - This is the manifest file which describes the **fundamental characteristics of the app** and defines each of its components.



## 11. AndroidManifest

- The 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.



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



- Default manifest file will look like as following file

```
<manifest
```

```
    xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.helloworld"
```

```
        android:versionCode="1"
```

```
        android:versionName="1.0" >
```

```
        <uses-sdk
```

```
            android:minSdkVersion="8"
```

```
            android:targetSdkVersion="22" />
```



# SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY  
**(DEEMED TO BE UNIVERSITY)**  
(Estd U/S(3) of UGC Act, 1956) Accredited with Grade "A" by NAAC



<application

    android:icon="@drawable/ic\_launcher"

    android:label="@string/app\_name"

    android:theme="@style/AppTheme" >

        <activity

            android:name=".MainActivity"

            android:label="@string/title\_activity\_main" >

                <intent-filter>

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

                />

                <category

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

                </intent-filter> </activity> </application> </manifest>



# AndroidManifest (con...)

- `<application>...</application>` tags enclosed the components related to the application.
- Attribute `android:icon` will point to the application icon available under `res/drawable-hdpi`.
- The `@string/app_name` refers to the `app_name` string defined in the `strings.xml` file, which is "HelloWorld"
- The `<activity>` tag is used to specify an activity and `android:name` attribute specifies the fully qualified class name of the `Activity` subclass.



## AndroidManifest (con...)

- The *android:label* attribute specifies a string to use as the label for the activity / application.
- The **action** for the intent filter is named *android.intent.action.MAIN* to indicate that this activity serves as the **entry point for the application**.
- The **category** for the intent-filter is named *android.intent.category.LAUNCHER* to indicate that the application can be launched from the device's launcher icon.



# AndroidManifest (con...)

- Following is the **list of tags which you will use in your manifest file** to specify different Android application components.
  - **<activity>** elements for activities
  - **<service>** elements for services
  - **<receiver>** elements for broadcast receivers
  - **<provider>** elements for content providers



# Practices

- To know about the history, features and various versions of Android
- Draw the Android architecture
- To study various tools used in Android development
- To study about Eclipse IDE
- To develop first Android App “Hello World”
- To implement the various Android layouts
- To implement the various Android UI controls
- To study the importance of Android application structure and Android manifest file