Mobile computing:- Mobile Computing refers a technology that allows transmission of data, voice and video via portable computing devices, such as smartphones, tablets, and laptops. It is free from having a connection with a fixed physical link.

Mobile Computing is a technology that provides an environment that enables users to transmit data from one device to another device without the use of any physical link or cables.

- Mobile communication
- Mobile Hardware
- Mobile Software

Application of mobile computing:

- Web or Internet access.
- Global Position System (GPS).
- Emergency services.
- Entertainment services.
- Educational services.

Fixed Network:- A fixed network typically refers to a telecommunications or data network infrastructure that is composed of stationary or immobile components, as opposed to mobile or wireless networks. It is also sometimes called a wired network because it relies on physical cables and connections to transmit data.

Here are some key concepts related to fixed networks:

- Physical Infrastructure
- Stability and Reliability
- High Bandwidth
- Security
- Cost of Deployment
- Coverage

Advantages	**Disadvantages**
o Reliability	- Limited Mobility
o High Bandwidth	- Infrastructure Costs
o Security	- Installation Complexity
o Consistent Speed	- Lack of Flexibility
o Lower Latency	- Vulnerability to Physical Damage
o Low Data loss rate	- Cable Management

Wireless Network:- A wireless network, also known as a Wi-Fi network, is a type of telecommunications or data network that transmits data without the use of physical cables.

Instead, it relies on wireless signals to connect devices to the internet or to communicate with each other

Multiplexing:- Multiplexing is a technique used to combine multiple data streams or signals into a single composite signal for transmission over a shared medium. It allows for the efficient use of available bandwidth and is commonly used in various communication systems, including telecommunication networks, broadcasting, and data transmission.

Spectrum:- In the context of mobile communication and telecommunications in general, the term "spectrum" refers to the range of electromagnetic frequencies that are used to transmit various types of signals, including voice, data, and other forms of information. This spectrum includes a wide range of frequencies, from very low frequencies used for AM radio to extremely high frequencies used for technologies like 5G.

Key component of spectrum :-

- Wireless Communication (Use Electronic waves)
- Frequency Bands
- Capacity and Speed
- Spectrum Management
- Range

Bluethoot technology:- Bluetooth is a short-range wireless technology used for connecting devices (e.g.,smartphones, headphones) wirelessly. It facilitates data and audio transmission over short distances, typically up to 100 meters.

Android Emulator:- The Android SDK provides a virtual mobile device emulator that runs on a computer. An android emulator is used for executing, debugging, and testing android applications. The emulator helps the developer to run a trial product virtually without an actual hardware device. An emulator is similar to a physical hardware mobile device including all the features that an actual mobile contains.

Feature or Advantages of AVD:-

- Transfer of files is faster in virtual devices, as here a file can be transferred through drag and drop.
- Emulator enables the programmer to work with physical sensors
- Android emulator enables you to choose any version of your choice
- The emulator is accessible on multiple platforms, including Windows, macOS, and Linux
- Developers can emulate a wide range of Android devices, screen sizes, resolutions, and hardware configurations
- Developers do not need to invest in physical Android devices, which can be costly

Dis Advantages of AVD:-

- Android emulator works with slow speed as compared to actual physical devices.
- Emulators cannot recognize the speed and performance of the battery, location, and hardware-related activities.
- Testing on an emulator is not accurate as you are using a virtual device not an actual device.
- One cannot identify issues related to networks or notifications on an emulator.
- The emulator crashes at launch if it does not receive enough disk space.

JAVA DEVLOPMENT KIT :- The JDK is a software package that contains a variety of tools and utilities that make it possible to develop, package, monitor and deploy applications that build for any standard Java platform, including Java Platform, Standard Edition (Java SE); Java Platform, Micro Edition (Java ME); and Java Platform, Enterprise Edition (Java EE).

The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) etc. to complete the development of a Java Application.

Android Studio:- Android Studio is the official integrated development environment (IDE) for Android app development. It is specifically designed to streamline the process of creating, testing, and deploying Android applications. Android Studio is developed by Google and is the most popular and widely used IDE for Android development

Features:-

- Android Emulator Integration
- User Interface (UI) Design Tools
- Auto code complete
- Version Control Integration
- Instant Run
- Kotlin Support
- Google Services Integration
- Regular Updates
- Official and Trusted

Project Structure of Android:-

- Android App modules
  - Manifests
  - Java
  - Res
- Library Modules
- Google App Engine Modules

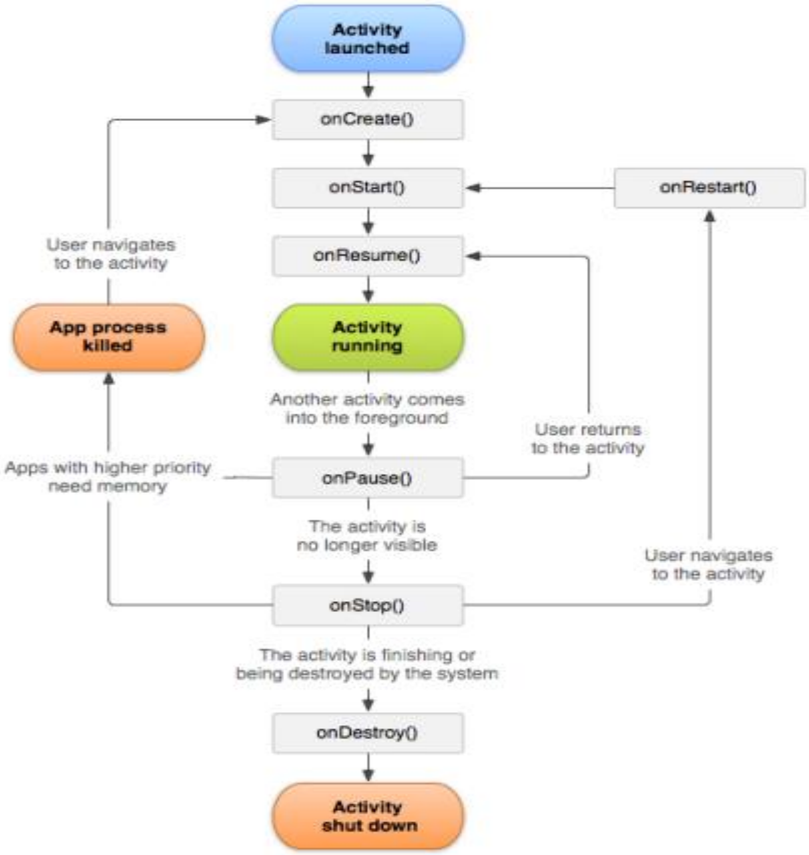
Android SDK Manager:- The Android SDK Manager is a graphical user interface (GUI) tool that is part of the Android Software Development Kit (SDK). It provides developers with a convenient way to manage various components, tools, and resources required for Android app development.

Tool -> Android -> SDK Manager

Activity:- An Android Activity is a fundamental component of the Android application architecture. It represents a single screen with a user interface (UI) that the user can interact with. Activities play a crucial role in the user's interaction with your Android app. They are responsible for displaying the user interface, handling user input, and managing the lifecycle of the app's user interface.

In short Activity is a single Frame of UI.

Lifecycle of Activity And it's Methods :-


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Layout:- In the context of Android app development, a "layout" refers to the visual arrangement of user interface (UI) elements within an Activity or Fragment. Layouts define how the various widgets and components such as buttons, text views, images, and input fields are displayed on the screen, and how they are organized relative to each other. Layouts are typically defined using XML files, and they play a crucial role in designing the user interface of an Android application.

In your Activity or Fragment code, you can "inflate" a layout by using the **setContentView()** method (for Activities) or by inflating it with a **LayoutInflater** (for Fragments).

MORE

XML:- XML stands for Extensible Markup Language. It is a text-based markup language derived from Standard Generalized Markup Language (SGML).

Syntax Rules for XML Declaration:-

- The XML declaration is case sensitive and must begin with "" where "xml" is written in lower-case.
- it strictly needs to be the first statement of the XML document.
- An HTTP protocol can override the value of encoding that you put in the XML declaration.

Syntax Rules for Tags and Elements

- Nesting of Elements (Last open First close)
- Root Element (Only one root)
- Case Sensitivity

characteristics of XML:-

- Self-Descriptive
- Hierarchical Structure
- Extensible
- Human-Readable

USE OF XML:-

- Web Development
- Data Interchange
- Document Markup
- Data Storage and Exchange
- Publishing and Content Management
- Database Data Export/Import
- Mobile Application Configuration
- Financial and Banking
- Healthcare
- Aeronautics and Aerospace

Markup Language:- a markup language is a set of symbols that can be placed in the text of a document to demarcate and label the parts of that document.

XML References:- References usually allow you to add or include additional text or markup in an XML document. References always begin with the symbol "&" which is a reserved character and end with the symbol ";".

Types of References:-

(1) Entity References
(2) Character References

XML Document Structure:-

1) Document Prolog Section (Contain XML Declaretion)
2) Document element section

Core Building Blocks of Android:-

- Activity
- View
- Intent
- Content Provider
- Fragment
- Android Mainfest.xml
- AVD (Android Virtual Device

R.Java File:- R.java is a generated Java file that is used in Android app development. It is automatically created by the Android build system during the compilation of your Android application and is located in the "gen" (short for "generated") folder of your project.

Components of R.Java:-

- Resource ID
- Auto-Generated
- Regenerated on Changes
- Accessing Resources

Android Mainfest:- *The **AndroidManifest.xml file*** contains information of your package, including components of the application such as activities, services, broadcast receivers, content providers etc.

Component:-

- <manifest>
- <application>
- <activity>
- <intent-filter>
- <Action>
- <category>

Android Hide Title:- requestWindowFeature(Window.FEATURE_NO_TITLE)

Hide Title Bar:- getSupportActionBar().hide();

Hiding Title Bar:-

1) Using the 'NoTitleBar' Theme:

In Manifestfile.xml

```
<activity
   android:name=".YourActivity"
   android:theme="@android:style/Theme.NoTitleBar">
   <!-- ... -->
</activity>
```

2) Programmatically hiding the title bar:

```
requestWindowFeature(Window.FEATURE_NO_TITLE); // Before calling setContentView
getSupportActionBar().hide();

setContentView(R.layout.your_layout);
```

3) Using 'android:windowFullscreen' attribute:

Write this in your style.xml

```
<style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <item name="android:windowFullscreen">true</item>
</style>
```

And then apply this theme to your activity in the AndroidManifest.xml file:

```
<activity

        android:name=".YourActivity"

        android:theme="@style/AppTheme">

        <!-- ... -->

</activity>
```

screen Orientation (Portrait, Landscape):- In Android, screen orientation refers to the orientation in which the device's screen is currently displayed. The two primary screen orientations are:

1) Portrait Orientation
2) Landscape Orientation

1) Portrait Orientation:

- Portrait orientation is the default orientation for most Android devices.
- In portrait mode, the screen is taller than it is wide.
- It's often used for displaying content in a vertical layout, such as reading text or using messaging apps.
- The orientation is represented by the value ActivityInfo.SCREEN_ORIENTATION_PORTRAIT.

2) Landscape Orientation:

- Landscape orientation is when the screen is wider than it is tall.
- It's commonly used for displaying content horizontally, like watching videos, playing games, or viewing maps.
- The orientation is represented by the value ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE.

Way to Orientation:-

1) Manifest File:

You can set the default screen orientation for your entire app or specific activities in the AndroidManifest.xml file. You can use the android:screenOrientation attribute to specify the desired orientation. For example**:**

```
<activity android:name=".YourActivity"
      android:screenOrientation="portrait">
  <!-- ... -->
</activity>
```

2) **Programmatically:**
You can also control screen orientation programmatically within your app. To lock the screen orientation to portrait or landscape, you can use the following code in your activity:

For Portrait orientation:

```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);
```

For Landscape orientation:

```
setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE);
```

Toast:- a Toast is a simple pop-up message that appears briefly on the screen to provide users with short, non-intrusive notifications or information. Toasts are typically used for displaying messages, warnings, or brief feedback to the user.

```
Toast.makeText(context, "Trial Toast MSG ", Toast.LENGTH_SHORT).show();
```

Toast Class:- The Toast class in Android is used to display short-duration messages or notifications to the user. These messages are typically displayed as pop-up windows that briefly appear on the screen and then disappear after a short period of time. Toasts are often used to provide users with quick feedback or information that doesn't require any user interaction. The Toast class is part of the Android SDK and is commonly used for displaying brief messages in applications.

ToggleButton:- ToggleButton is used to allow users to perform two operations on a single button. These are used to perform on and off operations like that of Switch. ToggleButton can perform two different operations on clicking on it. In this article, we will take a look at How to implement ToggleButton in Android.

Attributes of toggle Button:-

- textOff

- textOn

Event methods :

- getTextOff()
- getTextOn()
- setChecked()