SRKR ENGINEERING COLLGE(A)

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

II/IV-B.TECH-II SEMESTER (2023-2024)

MADK LAB (B20AM2205)

Name: **Regd. No: 22B91A61B0**

S.No	Date	Description	Sign
1		Introduction to mobile technologies and devices, Android platform and applications overview	
2		Setting Android development environments	
3		Writing Android applications, Understanding anatomy of an Android application	
4		Develop an application that uses GUI components ,Fonts and Colours	
5		Develop an application that uses Layout Managers and event listeners	
6		Write an application that draws basic graphical primitives on the screen.	
7		Create a quadrant in app and insert different objects.	
8		Create an app that displays multiple images in a page in vertical order	
9		Create an app that has multiple surfaces	
10		Implement login activity page	
11		Implement an application that creates an alert upon receiving a message	

RECORD MARKS :

Signature of the Faculty

Exercise-1:

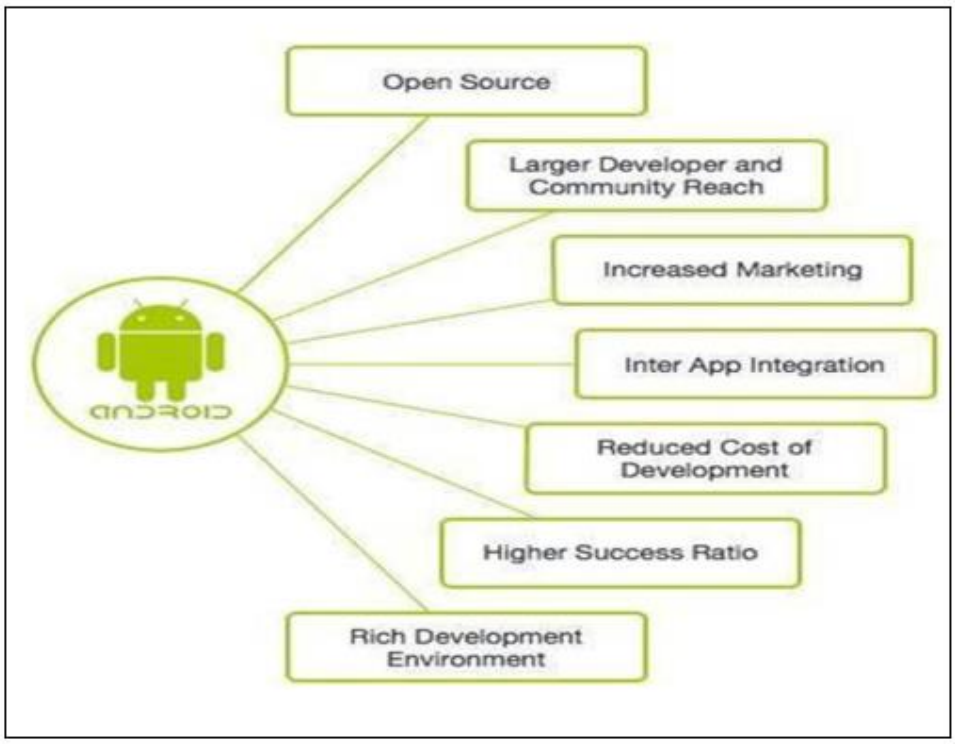
Aim: Introduction to mobile technologies and devices, Android platform and applications overview

Program:

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 runon different devices powered by Android.

Why Android?



Features of Android

Android is a powerful operating system competing with Apple 4GS and supports great features.Few of them are listed below …

Sr.No.	Feature & Description
1	**Beautiful UI** Android OS basic screen provides a beautiful and intuitive user interface.
2	**Connectivity** GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.

3	**Storage** SQLite, a lightweight relational database, is used for data storage purposes.
4	**Media support** H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, OggVorbis, WAV, JPEG, PNG, GIF, and BMP.
5	**Messaging** SMS and MMS
6	**Web browser** Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScriptengine supporting HTML5 and CSS3.
7	**Multi-touch** Android has native support for multi-touch which was initially made available in handsetssuch as the HTC Hero.
8	**Multi-tasking** User can jump from one task to another and same time various application can runsimultaneously.
9	**Resizable widgets** Widgets are resizable, so users can expand them to show more content or shrink them tosave space.
10	**Multi-Language** Supports single direction and bi-directional text.
11	**GCM** Google Cloud Messaging (GCM) is a service that lets developers send short message datato their users on Android devices, without needing a proprietary sync solution.
12	**Wi-Fi Direct** A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peerconnection.
13	**Android Beam** A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

History of 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. The code names of android ranges from A to N currently, such as Aestro, Blender, Cupcake, Donut, Eclair, Froyo, Gingerbread, Honeycomb, Ice Cream Sandwitch, Jelly Bean, KitKat, Lollipop and Marshmallow.

What is API level?

API Level is an integer value that uniquely identifies the framework API revision offered by aversion of the Android platform.

Platform Version	API Level	VERSION_CODE	
Android 6.0	23	MARSHMALLOW	
Android 5.1	22	LOLLIPOP_MR1	
Android 5.0	21	LOLLIPOP	
Android 4.4W	20	KITKAT_WATCH	
Android 4.4	19	KITKAT	
Android 4.3	18	JELLY_BEAN_MR2	
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	
Android 4.1, 4.1.1	16	JELLY_BEAN	
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	
Android 3.0.x	11	HONEYCOMB	
Android 2.3.4	10	GINGERBREAD_MR1	

Android 2.3.3			
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD	
Android 2.2.x	8	FROYO	
Android 2.1.x	7	ECLAIR_MR1	
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	

Exercise-2:

Aim : Setting Android development environments

Program: Android Studio is one of the best programs for Android app development. It offers a plethora of features and integrated tools for creating professional Android apps.

How to Download and Install the Java JDK

Before installing Android Studio, you first need to install the Java JDK and set up the Java environment to start working on it.

The Java Development Kit (JDK) is a required development package for creating Java-based applications. Additionally, it also includes tools for developers for testing Java-coded applications.

Follow these steps to get the latest version of Java JDK for Windows and set it up:

1. Go to the official Oracle Java Downloads page.
2. Choose the most recent version of Java JDK and download the x64

MSI Installerunder the Windows tab.

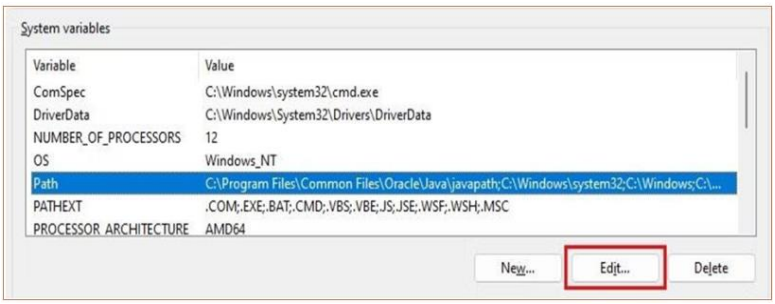
1. Open the file you've just downloaded from the Downloads folder.
2. Follow the on-screen instructions to install the latest Java JDK.

Now that you've successfully installed JDK, you need to set up a Java environment so that Android Studio can easily find the location of Java on your desktop.

Setting Up the Java Environment

Here are the steps you've to perform for setting up the Java environment:

1. Open Windows search and type "Edit environment variables."
2. Hit the Enter key to open the Environment Variables window.
3. Now, under the System variables section, click Path > Edit > New



4. Copy the path of the folder where the JDK binaries are installed. For example, in our case, the path is C:\Program Files\Java\jdk-18.0.1.1\bin

5. Switch to the Environment variables window and paste the path you've copied and save the changes.
6. Next, under the User Variables section, click New.

7. Add PATH_HOME in the variable name box and C:\Program Files\Java\jdk-18.0.1.1 in the variable value box.

Finally, save the changes by clicking OK.

How to Install Android Studio on Windows

Follow these steps to download and install Android Studio for Windows:

1. Visit the Android Developers website and click on Download Android Studio.
2. Now, open the downloaded file to install Android Studio.
3. Click Next to continue installing Android Studio.
4. Further, click Install and wait while Android Studio is being installed on your PC.
5. Once done, click Finish to close the installer and start Android Studio.
6. Android Studio will now ask you to import settings. Click OK to continueto set up Android Studio.

The System Requirements for Android Studio on Windows

Your PC must meet the following requirements to run Android Studio efficiently:

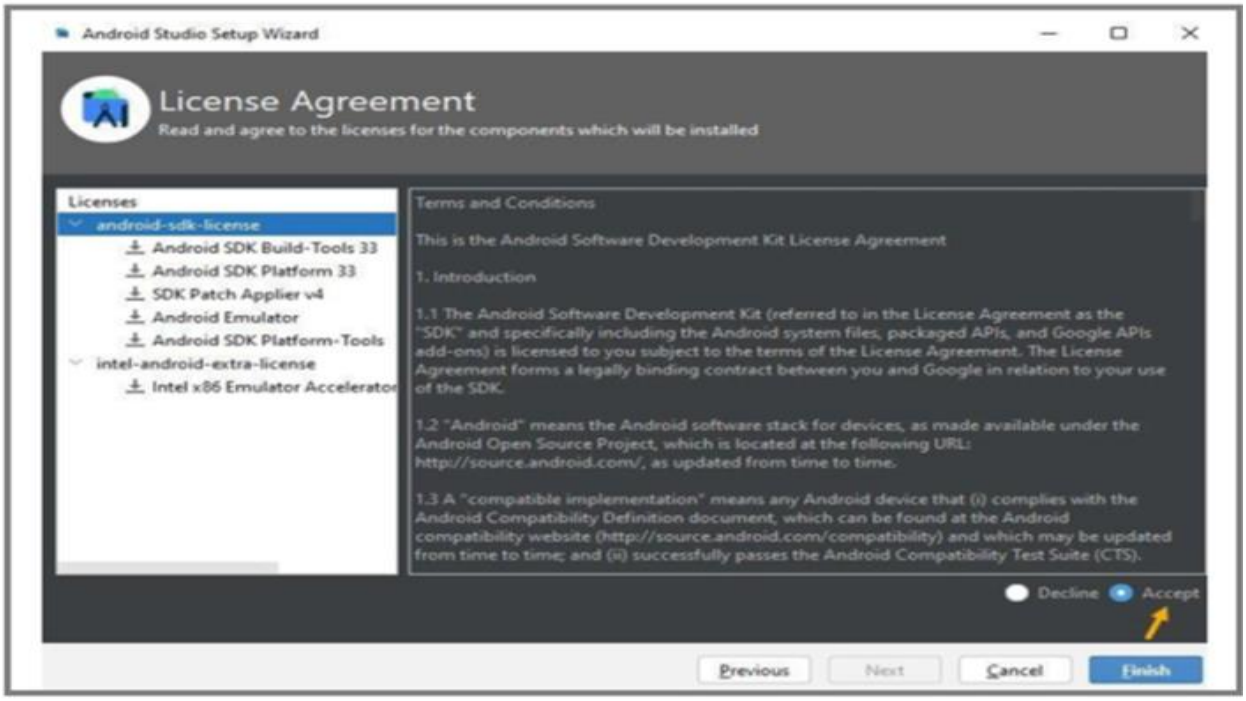
1. 64-bit Microsoft Windows OS 8/10/11.
2. 1280 x 800 minimum screen resolution.
3. x86_64 CPU architecture; 2nd generation Intel Core or newer; or AMD CPU with Hypervisorsupport.
4. 8GB of RAM or more.
5. 12GB of available disc space is required (IDE + Android SDK + Android Emulator).

How to Set Up Android Studio on Windows

While you've installed Android Studio on your Windows PC, to create a new project you'll have to configure the Setup Wizard. The Setup Wizard will help you create a development environment for your future projects.

Follow these steps to set up Android Studio:

1. In the Setup Wizard window, select the Standard option to apply the recommendedsettings and click Next.
2. Next, select a theme you want and click Next again to verify the settings you'veconfigured.
3. On the following screen, select Accept and click Finish to apply the changes.



4. Wait till the Setup Wizard downloads some required files. It may take somewherearound 15 to 30 minutes, depending on your network speed.
5. Once the download is completed, click Finish to complete the setup process

Exercise-3:

Aim: **Writing Android applications, Understanding anatomy of an Android applications.**

Program:

There are four building blocks to an Android application:

r.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

Not every application needs to have all four, but your application will be written with some combination of these.

Once you have decided what components you need for your application, you should list them in a file called AndroidManifest.xml. This is an XML file where you declare the components of your application and what their capabilities and requirements are.

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

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

```
public class MyReceiver extends BroadcastReceiver {
  public void onReceive(context,intent){}
}
```

Content Providers

A content provider component supplies data from one application to others on request. Such requests are handled by the methods of the *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(){}
}
```

We will go through these tags in detail while covering application components in individualchapters.

Additional Components

There are additional components which will be used in the construction of above mentionedentities, their logic, and wiring between them. These components are −

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