



Silver Oak College of Engineering and Technology

GUJARAT TECHNOLOGICAL UNIVERSITY BACHELOR OF ENGINEERING

Mobile Application Development
(3161612)
6th SEMESTER

INFORMATION TECHNOLOGY

Laboratory Manual



DEPARTMENT OF INFORMATION ENGINEERING

VISION

To be recognized for the quality education and research in the field of Information Technology known for its accomplished graduates.

MISSION

1. Continually improve the standard of our graduates by engaging in innovative teaching learning methods with high caliber motivated faculty members keeping in-line with the rapid technological advancements.
2. Promote and support research activities over a wide range of academic interests among students and staff for growth of individual knowledge and continuous learning.
3. Provide an education system that promotes innovation, creativity, entrepreneurial spirit, leadership as well as freedom of thought with emphasis on professionalism and ethical behavior.

PROGRAM EDUCATIONAL OBJECTIVES (PEO):

PEO1: To provide fundamental knowledge of science and engineering for an IT professional and to equip them with proficiency of mathematical foundations and algorithmic principles and inculcate competent problem-solving ability.

PEO2: To implant ability in creativity & design of IT systems and transmit knowledge and skills to analyze, design, test and implement various software applications.

PEO3: To exhibit leadership capability, triggering social and economic commitment and inculcate community services.

PEO4: To inculcate professional-social ethics, teamwork in students and acquaint them with requisite technical and managerial skills to attain a successful career.



PROGRAM OUTCOMES (POs):

Engineering Graduates will be able to:

- 1. Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2. Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3. Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4. Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5. Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
- 6. The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7. Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8. Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9. Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10. Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.



- 11. Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12. Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.



ARTIFICIAL INTELLIGENCE PRACTICAL BOOK
DEPARTMENT OF INFORMATION TECHNOLOGY

PREFACE

It gives us immense pleasure to present the first edition of Mobile Application Development Practical Book for the B.E. 3rd year students of Silver Oak College of Engineering and Technology.

The theory and laboratory course of Mobile Application Development, at Silver Oak College of Engineering and Technology, Ahmedabad, is designed in such a manner that students can develop the basic understanding of the subject during theory classes and gain the hands-on practical experience during their laboratory sessions.

The Laboratory Manual presented here to you help you in understanding various search methods, which explores the possible moves that one can make in a space of 'states', called the search space. It also takes you in learning various Practical of Android Application Development. It will help you in learning Android Studio, Eclipse and other software that are used in developing of Android Application.

Lab Manual Revised by: Prof. Bhavin Trivedi, Silver Oak College of Engineering and Technology

Prof. A. R Kazi, Aditya Silver Oak Institute of Technology

Lab Manual Revision No.: SOGI_3161612_LM_2020-21_1



INSTRUCTIONS TO STUDENTS

1. Be prompt in arriving to the laboratory and always come well prepared for the experiment.
2. Students need to maintain a proper decorum in the computer lab. Students must use the equipment with care. Any damage is caused is punishable.
3. Students are supposed to occupy the systems allotted to them and are not supposed to talk or make noise in the lab.
4. Students are required to carry their observation book and lab records with completed exercises while entering the lab.
5. Lab records need to be submitted every week.
6. Students are not supposed to use pen drives in the lab.
7. The grades for the Artificial Intelligence Practical course work will be awarded based on your performance in the laboratory, regularity, recording of experiments in the Artificial Intelligence practical Final Notebook, lab quiz, regular viva-voce and end-term examination.
8. Find the answers of all the questions mentioned under the section 'Post Practical Questions' at the end of each experiment in the Artificial Intelligence Practical Book.



CERTIFICATE

This is to certify that

Mr./Ms..... with
enrolment no.from Semester.....Div.....has
successfully completed his/her laboratory experiments in the **MOBILE
APPLICATION DEVELOPMENT (3161612)** from the department of
.....during the academic year..... -.....

Date of Submission:

Staff Incharge:

Head of Department:



TABLE OF CONTENT

Sr. No	Experiment Title	Page No		Date of Start	Date of Completion	Sign	Marks (out of 10)
		To	From				
1	Android studio installation and study of architecture of Android.						
2	Create “Hello World” application. That will display “Hello World” in the middle of the screen using Text View Widget in the red color.						
3	Create android application that demonstrates the android activity life cycle.						
4	Design an android application for simple calculator.						
5	Create android application that demonstrates the concept of INTENT.						
6	Create android applications that demonstrate the concept of fragments.						
7	Create android applications that demonstrate the concept of menu with navigation.						
8	Develop an application demonstrating Internal Storage to store private data on the device memory.						
9	Develop an application for working with graphics and animation.						
10	Create an application that shows your current location.						
11	Create registration application to store data in database.						



Practical 1

AIM: - Android studio installation and study of architecture of Android.

Theory

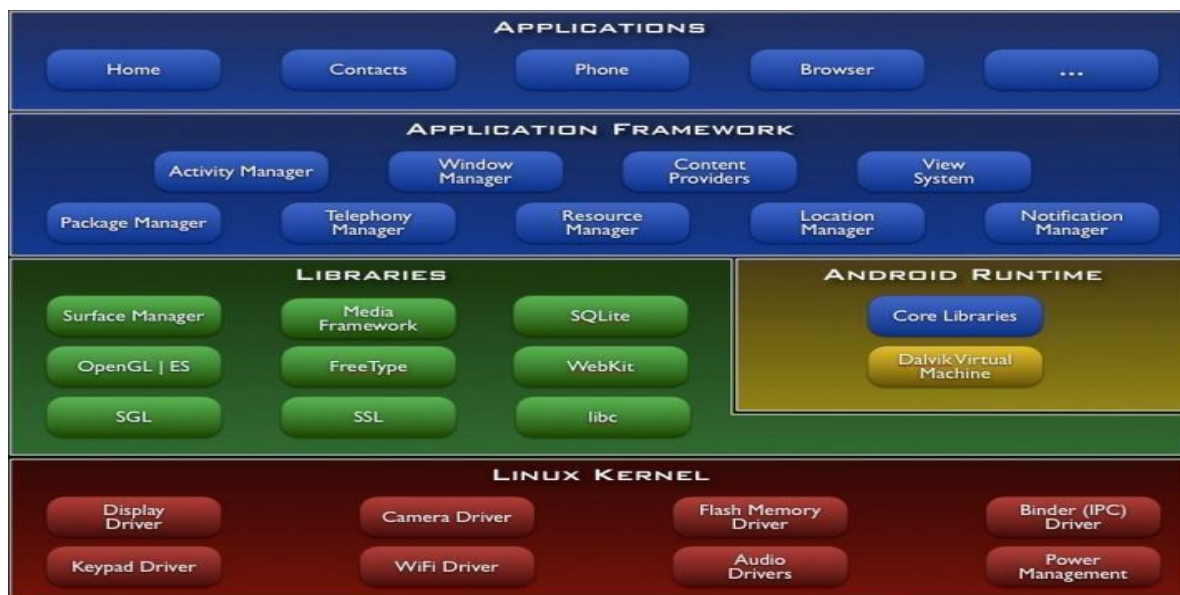
Android:

Android is a mobile operating system based on a modified version of the Linux kernel and other open source software, designed primarily for touch screen mobile devices such as smart phones and tablets. Android is developed by a consortium of developers known as the Open Handset Alliance, with the main contributor and commercial marketer being Google.

Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the [first commercial Android device](#) launched in September 2008. The current stable version is [Android 10](#), released on September 3, 2019.

Android Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.





Linux kernel

At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches.

This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

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 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 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.

Applications

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, and Games etc.

Android UI

An Android application user interface is everything that the user can see and interact with.

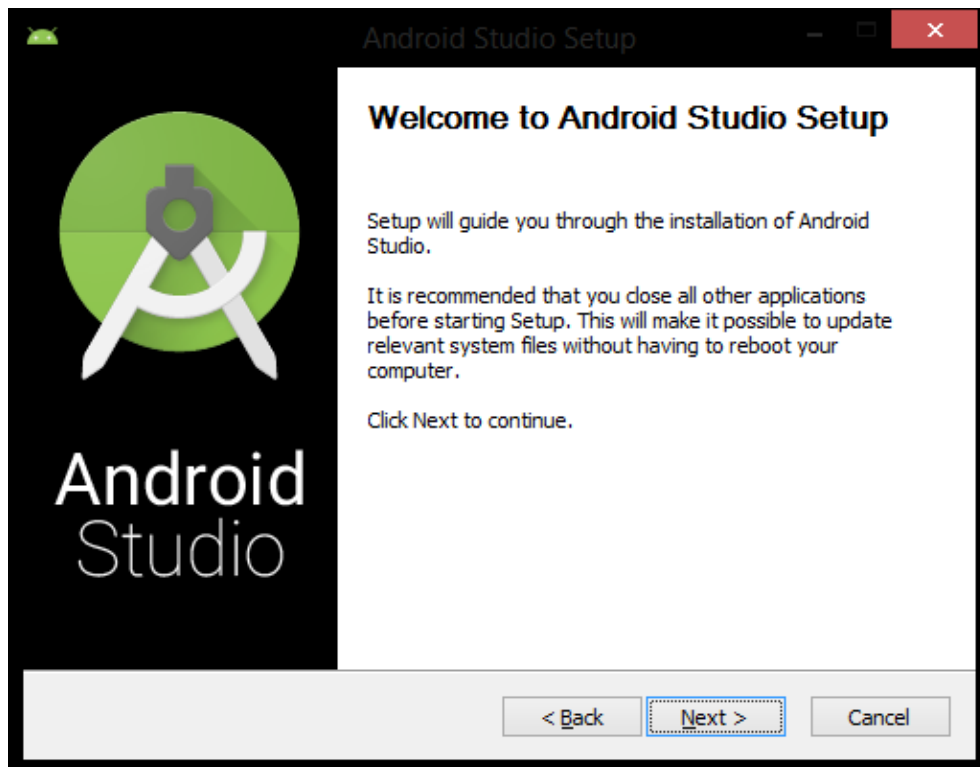
Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on [IntelliJ IDEA](#). On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

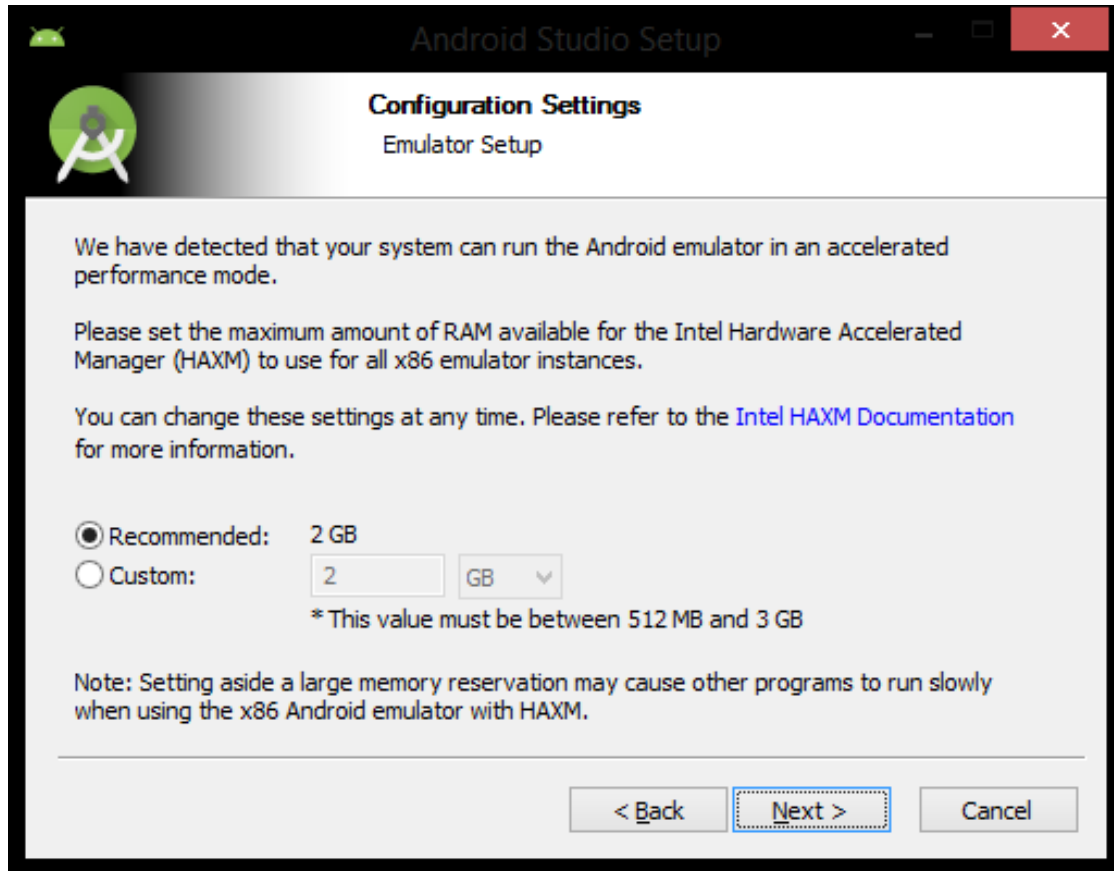
- A flexible Gradle-based build system. A fast and feature-rich emulator
- A unified environment where you can develop for all Android devices
- Apply Changes to push code and resource changes to your running app without restarting yourapp
- Code templates and GitHub integration to help you build common app features and importsample code. Extensive testing tools and frameworks
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK support



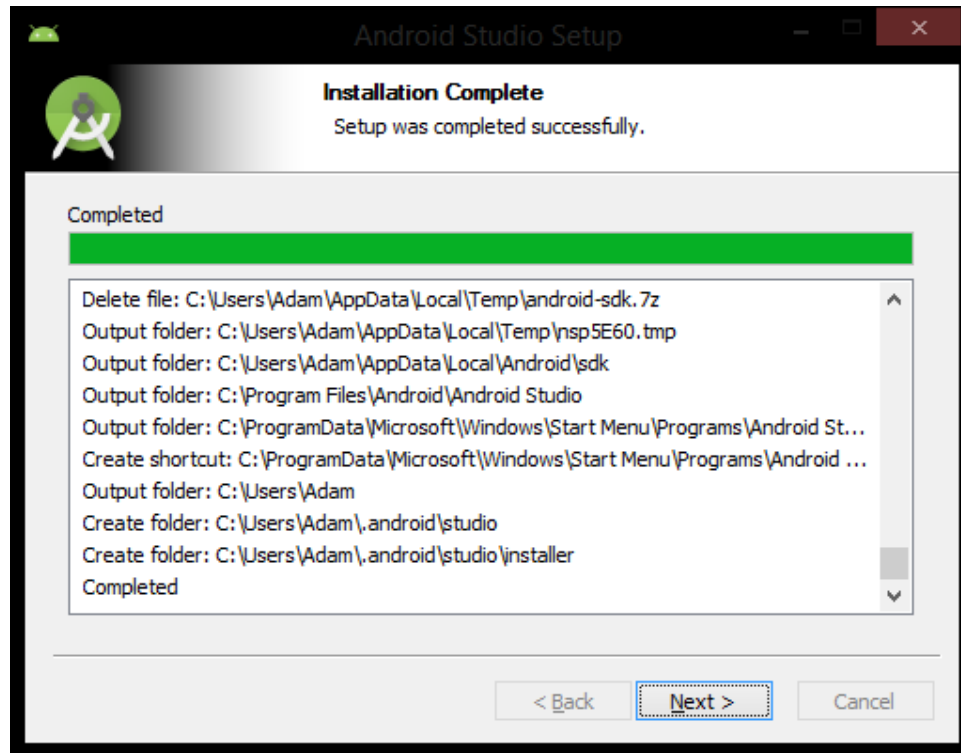
- Built-in support for [Google Cloud Platform](https://cloud.google.com/), making it easy to integrate Google Cloud Messaging and App Engine
1. Download Android Studio from <https://developer.android.com/studio/index.html> . Click on 'Download Android Studio'.
 2. Open the executable file android-studio-<xxx>.
 3. Once the setup loads, you will see the Welcome Screen



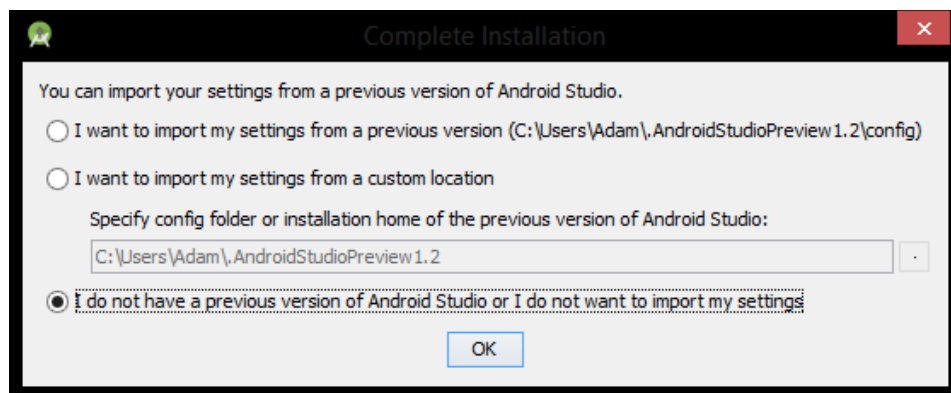
4. Click 'Next >' on the Welcome Screen.
5. When choosing components, ensure all of the checkboxes are checked in for each component to install. Once you are done, click 'Next >'.
6. Agree to the Android Studio and the Intel HAXM License Agreements after reading them.
7. Verify the install locations meet the installation requirements and click 'Next >'.
8. You may or may not see the emulator setup settings, just click 'Next >' after selecting the RAM size.



- 9 Finally, click 'Install'. You will see which operations are currently running in the installation process and a progress bar displaying their progress.
- 10 Once the installation process is finished click 'Next >'.



11. Android Studio is now set up. Check on 'Start Android Studio' and click 'Finish'.
12. You will see the Complete Installation screen below.
13. If you had a previous version of Android Studio installed prior, then check either the first or second radio box. Otherwise, check the last radio box and hit 'OK'.

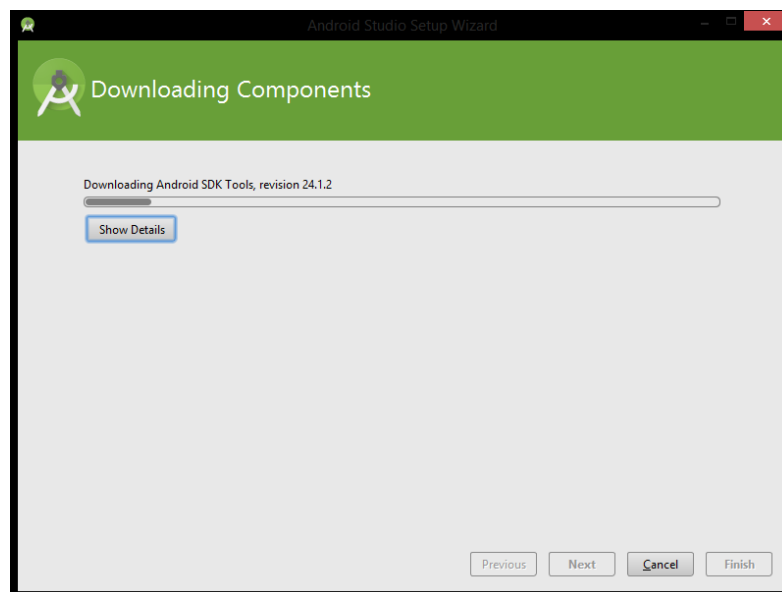




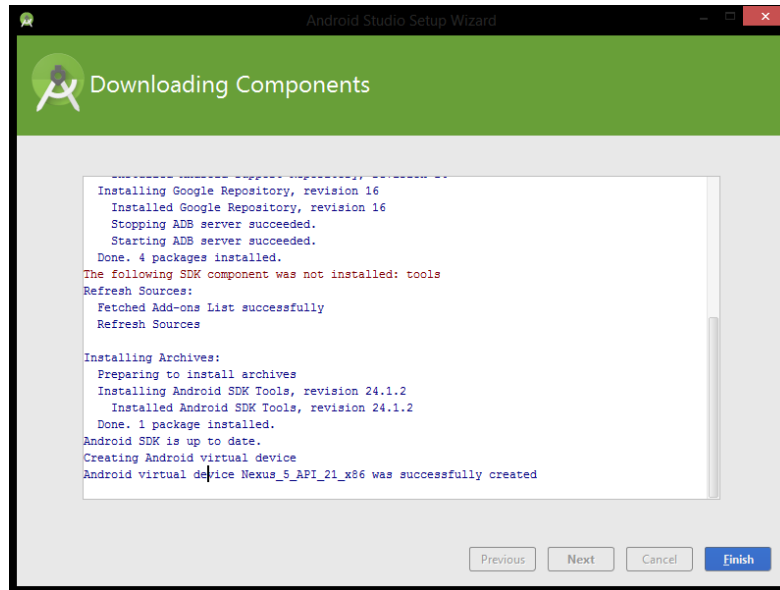
14. As Android Studio starts, the splash screen will appear.



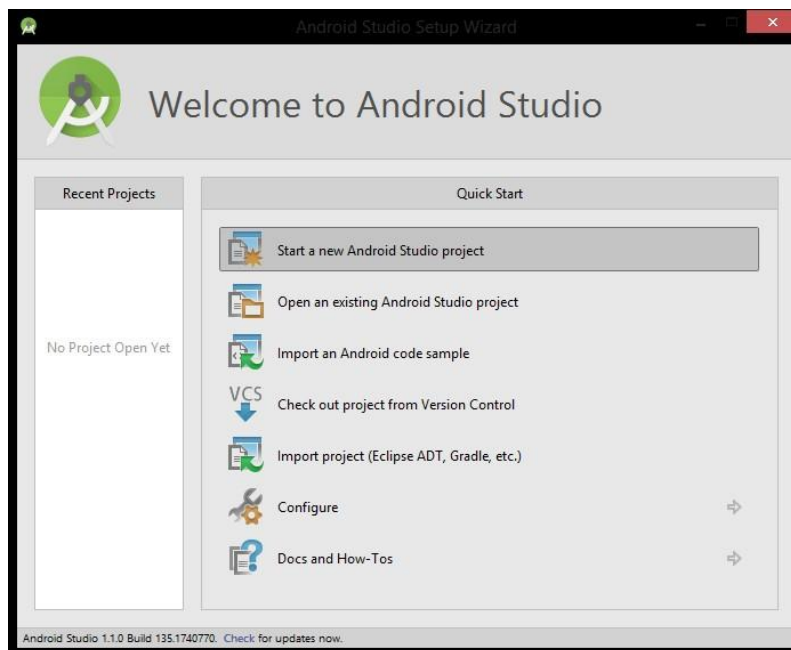
- a. After the splash screen you may see some additional setup operations run, such as downloading components.



- b. Once it is finished, click 'Finish'.



15. Welcome to Android Studio! In the next part we will start our first project





Conclusion

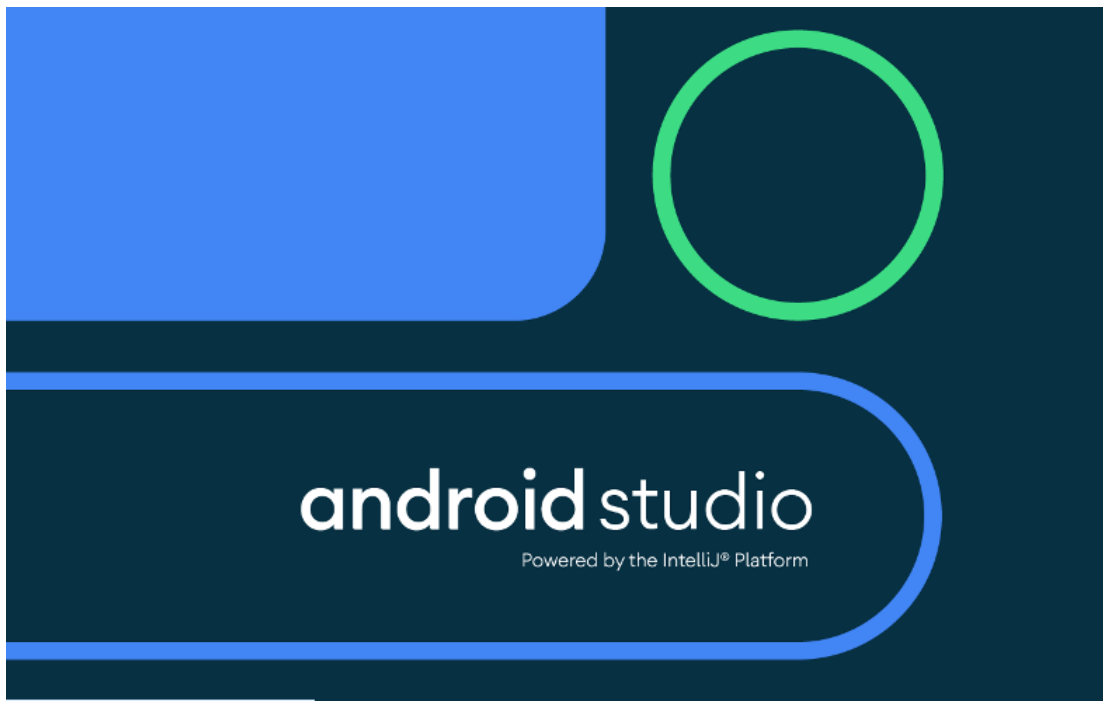


Practical 2

AIM: - Create “Hello World” application. That will display “Hello World” in the middle of the screen using Text View Widget in the red color.

Step 1: -

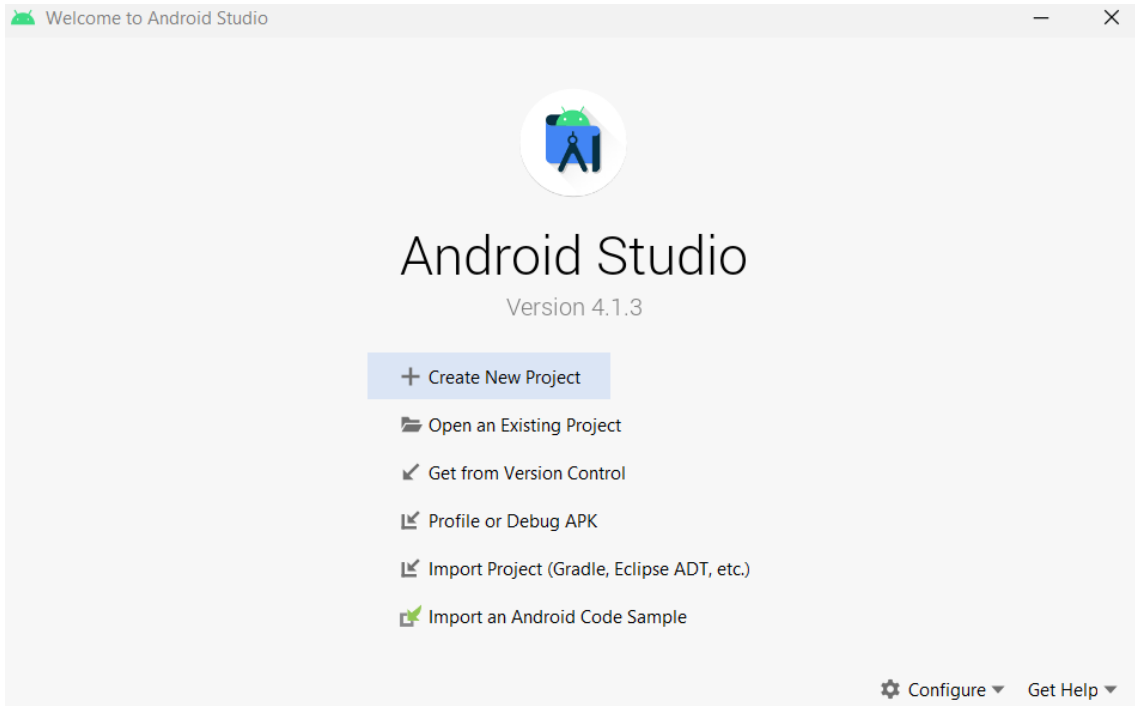
1. Open Android Studio application



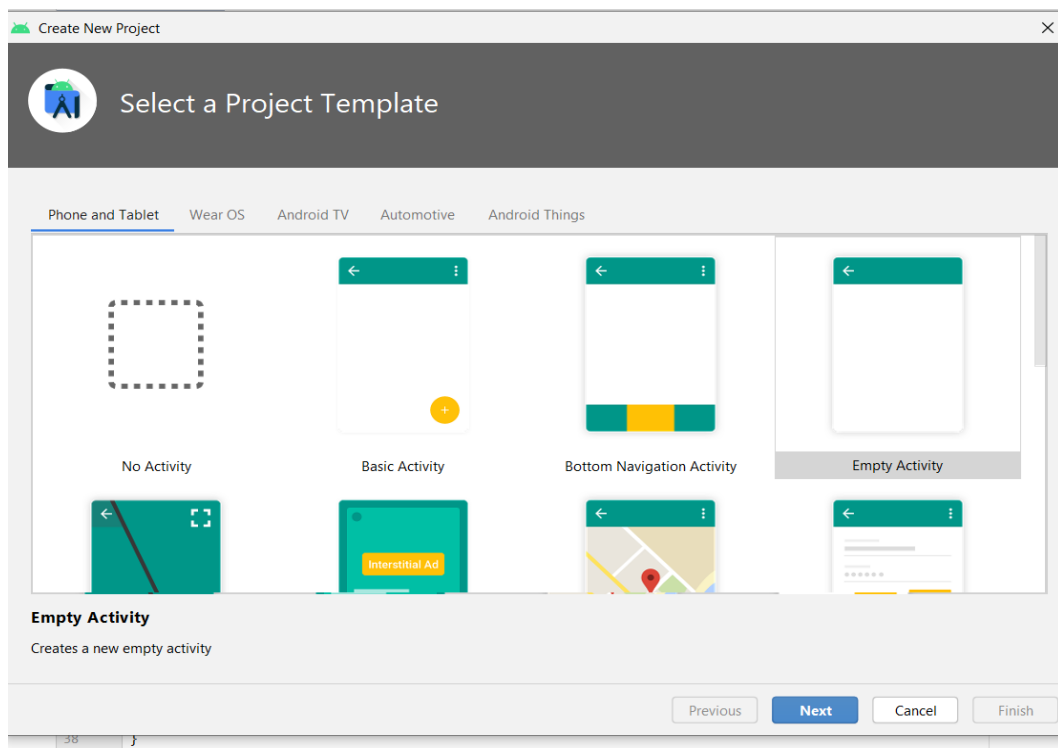
In this part you will create a simple Android application that displays the words, "Hello World!"



2. At the Welcome Screen, click on ‘Start a new Android Studio project’.

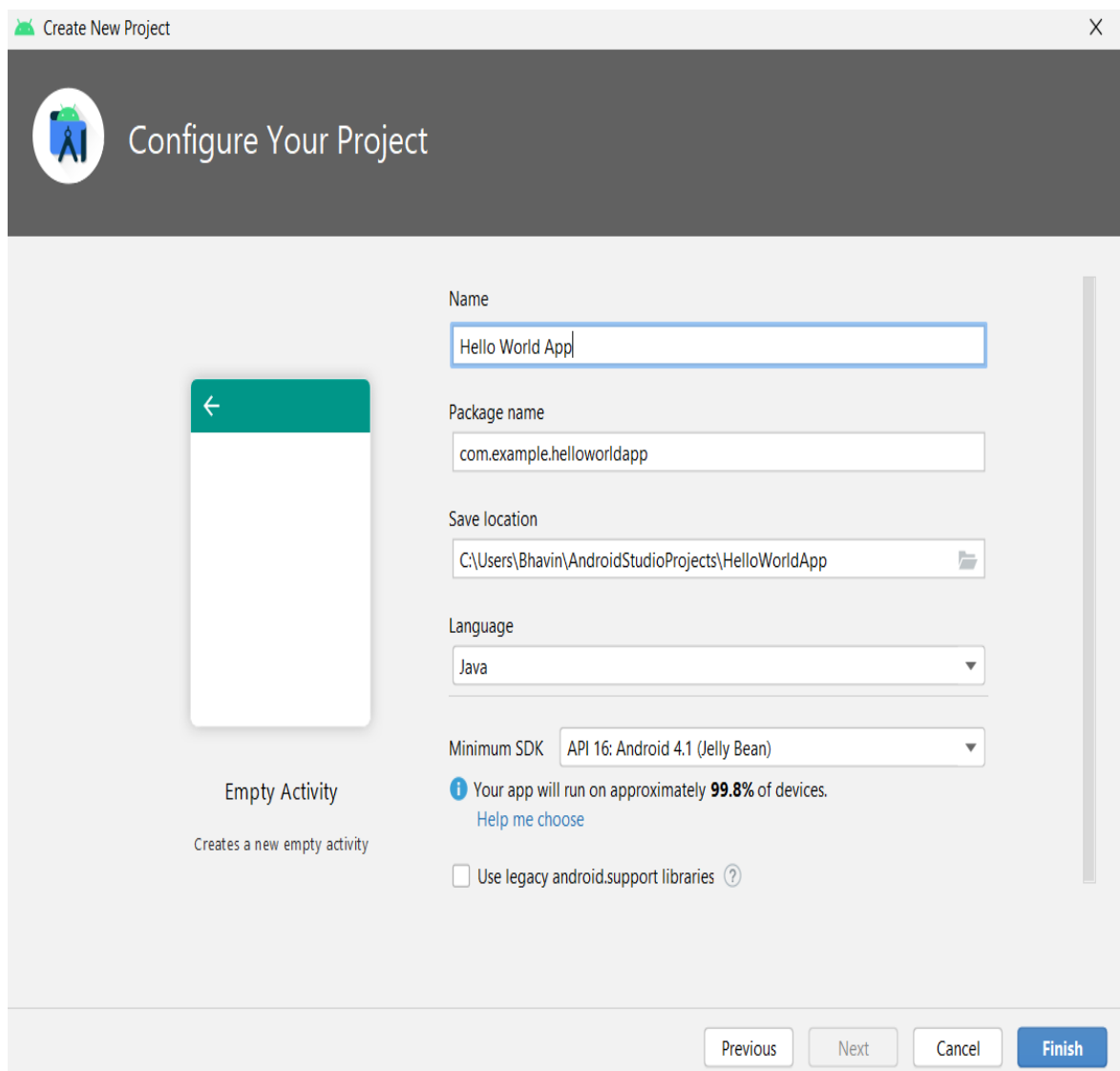


3. Select ‘Empty Activity’ and click ‘Next’





4. Enter the application name 'Hello World App' and note where the project is located. I am creating my project inside of my home directory.



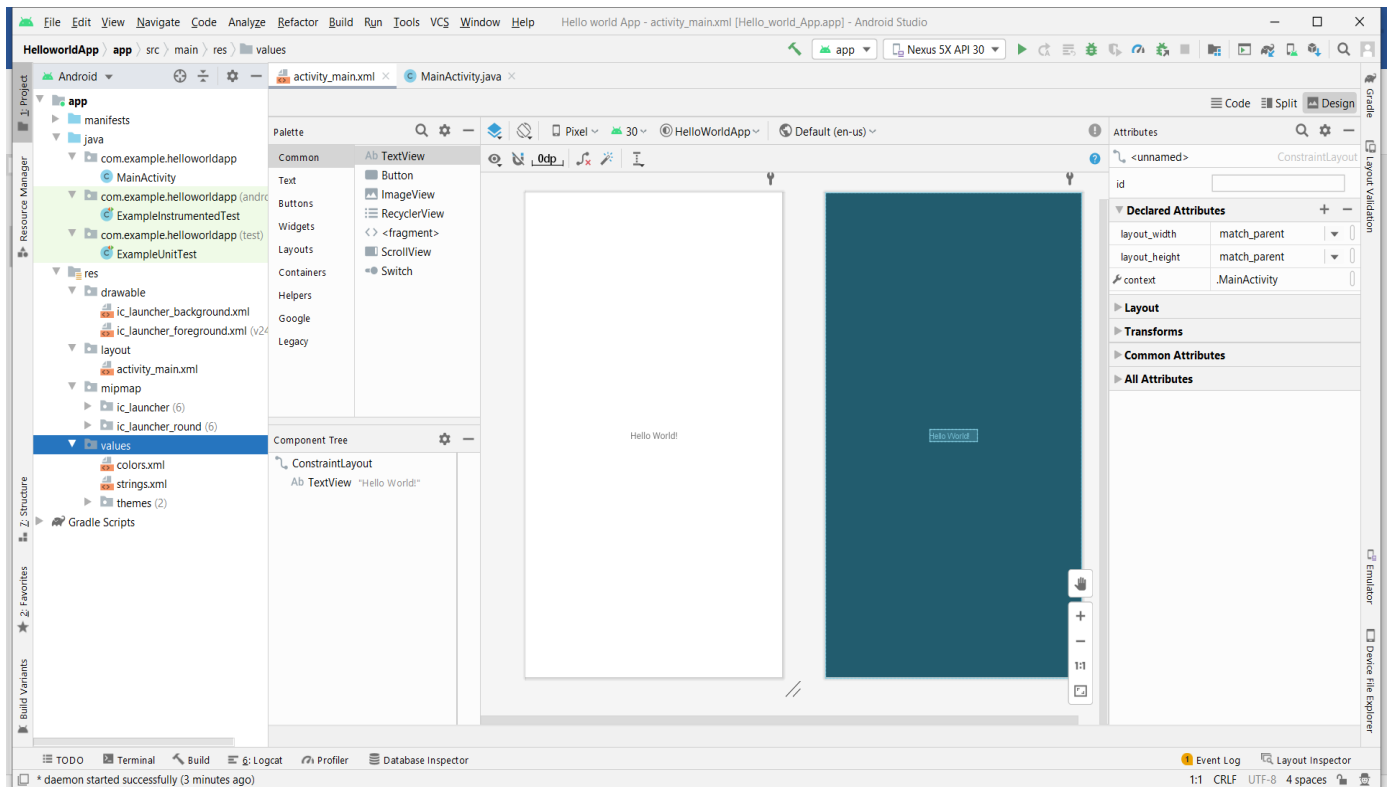
5. In the next window, leave all the settings as default, and then click Finish.
6. Android Studio will now create the project and build it.
7. Once the Android Studio IDE fully loads, click on the 'Related XML file' icon you will see by hovering over the icon in the middle of the window.



```

1 package com.example.helloworldapp;
2
3 import ...
4
5
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }

```





Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">

    <Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Open Activity 2"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



MainActivity.java



Output



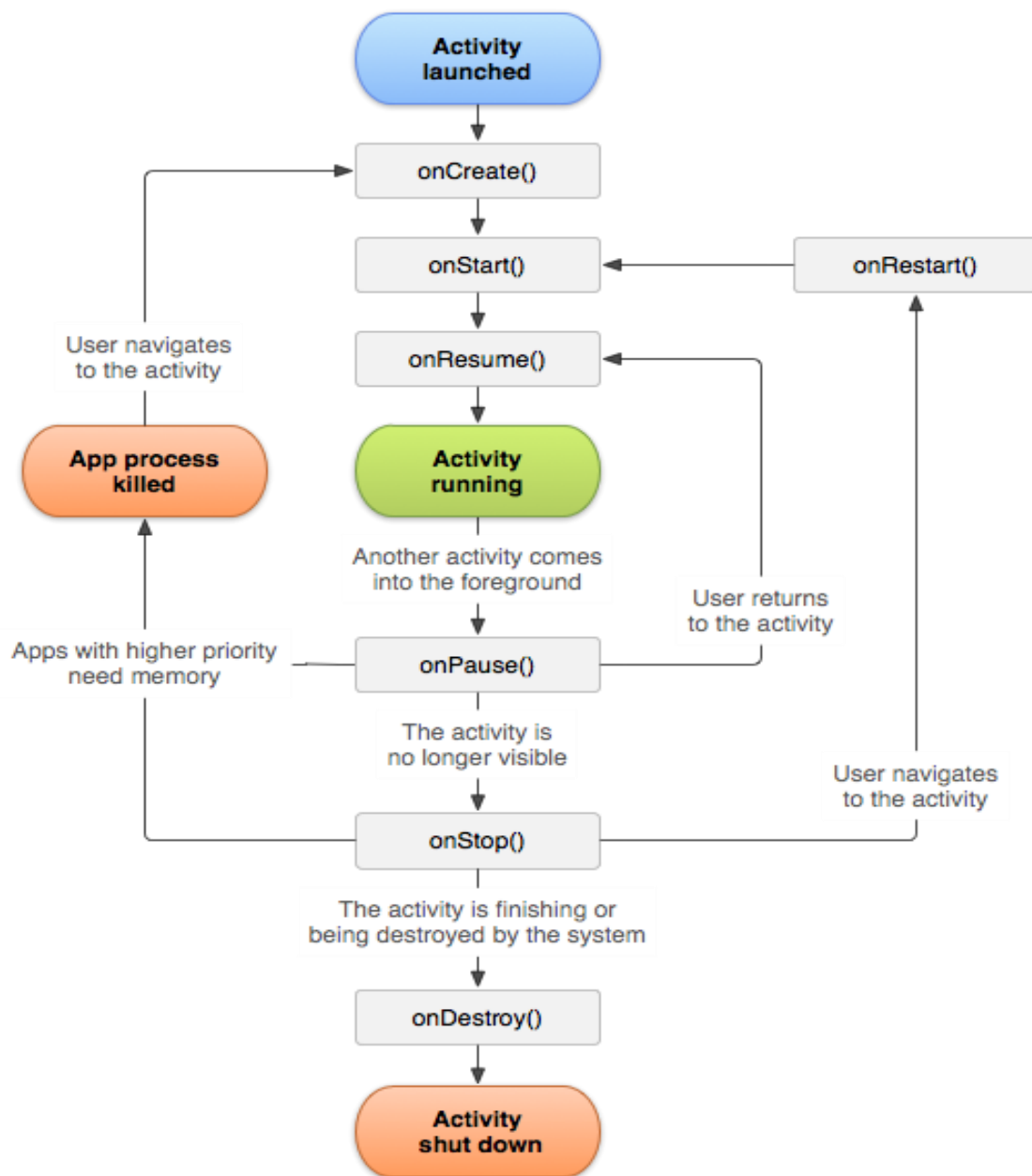
5. What is the difference between wrap_content and match_parent?

Conclusion

Practical 3

AIM: - Create android application that demonstrates the android activity life cycle.

To navigate transitions between stages of the activity lifecycle, the Activity class provides a core set of six callbacks: `onCreate()`, `onStart()`, `onResume()`, `onPause()`, `onStop()`, and `onDestroy()`. The system invokes each of these callbacks as an activity enters a new state.





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.

Activity_main.xml



MainActivity.java



Output

Practice

1. What's the difference between onCreate() and onStart()?
2. How does the activity respond when the user rotates the screen?



3. What's the difference between `onResume()` and `onPause()`?

4. Difference between Activity & Service

5. Choose the correct option regarding activity in android.
 - A. An activity is a window that contains the user interface of your application.
 - B. An application can have zero or more activities.
 - C. An application can have only one activity.
 - D. Option A and B are correct.

Conclusion



PRACTICAL 4

AIM: - Design an android application for simple calculator.

activity_main.xml





MainActivity.java





Output

Practice

- 1. What is View in Android?**
- 2. What View Group in Android?**



3. What are the containers?

4. What are the common layouts in Android?

5. What is the default Layout used by Android Studio?

Conclusion



PRACTICAL 5

AIM: - Create android application that demonstrates the concept of INTENT.

Intent

- ✓ Android Intent is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc.
- ✓ It is generally used with startActivity() method to invoke activity, broadcast receivers etc.
- ✓ The dictionary meaning of intent is **intention or purpose**. So, it can be described as the intention to do action.
- ✓ The LabeledIntent is the subclass of android.content.Intent class.

Android intents are mainly used to:

1. Start the service
2. Launch an activity
3. Display a web page
4. Display a list of contacts
5. Broadcast a message
6. Dial a phone call etc.

Types of Android Intents

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

1) Implicit Intent

- ✓ Implicit Intent doesn't specify the component. In such case, intent provides information of available components provided by the system that is to be invoked.
- ✓ For example, you may write the following code to view the webpage.



Example: -

```
Intent intent=new Intent(Intent.ACTION_VIEW);  
intent.setData(Uri.parse("http://www.javatpoint.com"));  
startActivity(intent);
```

2) Explicit Intent

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

Example: -

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

activity_main.xml



MainActivity.java



5. Explain the difference between implicit and explicit intent.

Conclusion



PRACTICAL 6

AIM: - Create android applications that demonstrate the concept of fragments.

Android Fragments

- ✓ Android Fragment is the part of activity, it is also known as sub-activity. There can be more than one fragment in an activity. Fragments represent multiple screen inside one activity.
- ✓ Android fragment lifecycle is affected by activity lifecycle because fragments are included in activity.
- ✓ Each fragment has its own life cycle methods that is affected by activity life cycle because fragments are embedded in activity.
- ✓ The `FragmentManager` class is responsible to make interaction between fragment objects.

Android Fragment Lifecycle

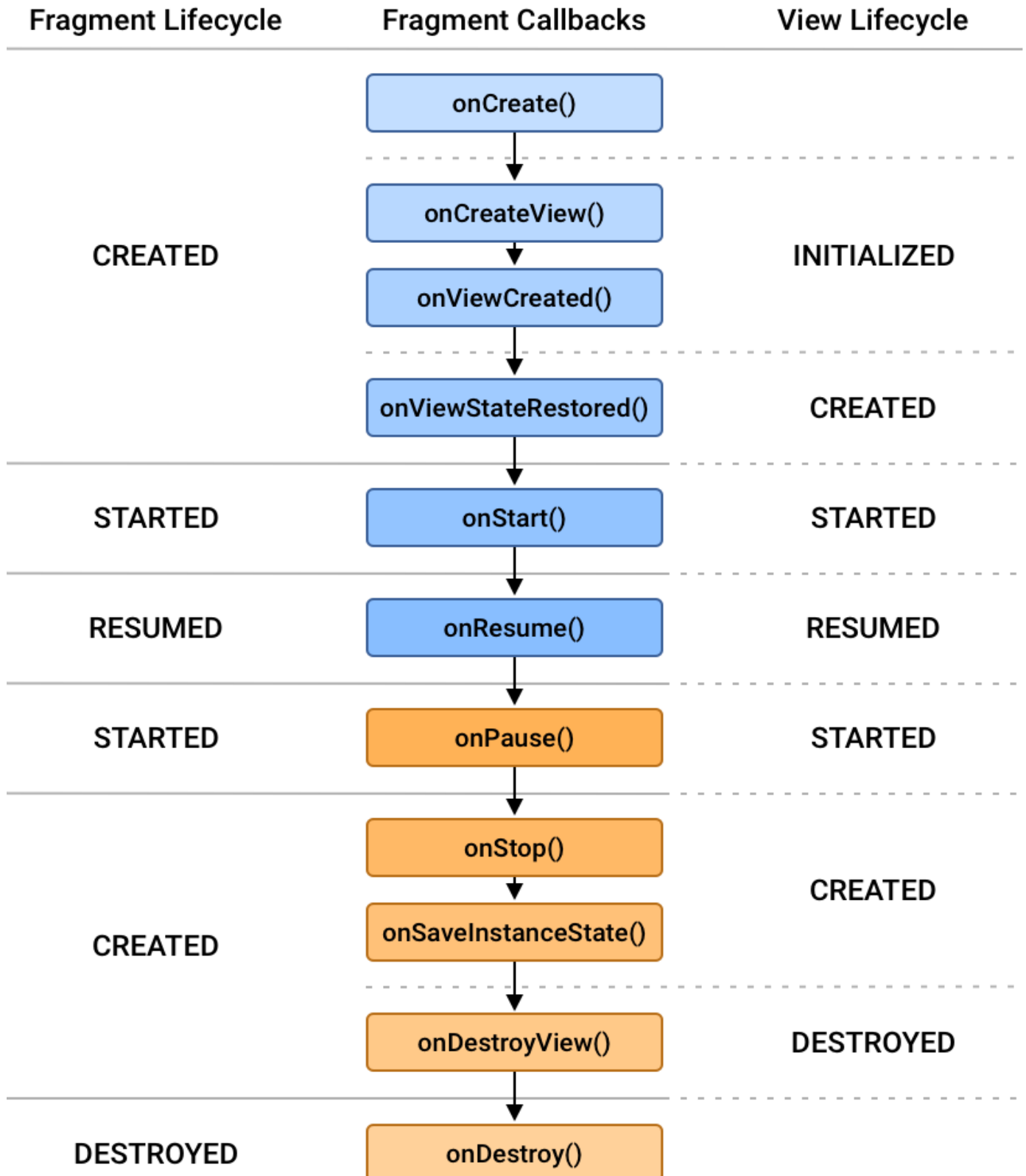
The lifecycle of android fragment is like the activity lifecycle. There are 12 lifecycle methods for fragment.

Each Fragment instance has its own lifecycle. When a user navigates and interacts with your app, your fragments transition through various states in their lifecycle as they are added, removed, and enter or exit the screen.

To manage lifecycle, `Fragment` implements `LifecycleOwner`, exposing a `Lifecycle` object that you can access through the `getLifecycle()` method.

Each possible Lifecycle state is represented in the `Lifecycle.State` enum.

1. `INITIALIZED`
2. `CREATED`
3. `STARTED`
4. `RESUMED`
5. `DESTROYED`





No.	Method	Description
1)	onAttach(Activity)	it is called only once when it is attached with activity.
2)	onCreate(Bundle)	It is used to initialize the fragment.
3)	onCreateView(LayoutInflater, ViewGroup, Bundle)	creates and returns view hierarchy.
4)	onActivityCreated(Bundle)	It is invoked after the completion of onCreate() method.
5)	onViewStateRestored(Bundle)	It provides information to the fragment that all the saved state of fragment view hierarchy has been restored.
6)	onStart()	makes the fragment visible.
7)	onResume()	makes the fragment interactive.
8)	onPause()	is called when fragment is no longer interactive.
9)	onStop()	is called when fragment is no longer visible.
10)	onDestroyView()	allows the fragment to clean up resources.
11)	onDestroy()	allows the fragment to do final clean up of fragment state.
12)	onDetach()	It is called immediately prior to the fragment no longer being associated with its activity.



activity_main.xml

MainActivity.java



Fragment1_main.xml

Fragment1.java



Fragment2_main.xml

Fragment2.java



OUTPUT



5. What are the event Listeners?

Conclusion



PRACTICAL 7

AIM: - Create android applications that demonstrate the concept of menu with navigation.

Navigation

Navigation refers to the interactions that allow users to navigate across, into, and back out from the different pieces of content within your app. Android Jetpack's Navigation component helps you implement navigation, from simple button clicks to more complex patterns, such as app bars and the navigation drawer. The Navigation component also ensures a consistent and predictable user experience by adhering to an established set of principles.

The Navigation component consists of three key parts that are described below:

Navigation graph: An XML resource that contains all navigation-related information in one centralized location. This includes all of the individual content areas within your app, called *destinations*, as well as the possible paths that a user can take through your app.

NavHost: An empty container that displays destinations from your navigation graph. The Navigation component contains a default NavHost implementation, NavHostFragment, that displays fragment destinations.

NavController: An object that manages app navigation within a NavHost. The NavController orchestrates the swapping of destination content in the NavHost as users move throughout your app.

Menu

Android Option Menus are the primary menus of android. They can be used for settings, search, delete item etc.

Here, we are going to see two examples of option menus. First, the simple option menus and second, options menus with images.

Here, we are inflating the menu by calling the `inflate()` method of `MenuInflater` class. To perform event handling on menu items, you need to override `onOptionsItemSelected()` method of `Activity` class.



activity_main.xml





MainActivity.java





Output

Practice

1. How to define Menu in XML File?



2. Android Different Types of Menus

3. Can you develop Android Apps with languages other than Java? If so, name some.

4. What is an Adapter?



5. What are the types of Android Adapters?

Conclusion



PRACTICAL 8

AIM: - Develop an application demonstrating Internal Storage to store private data on the device memory.

activity_main.xml



MainActivity.java





Output

Practice

- 1. What are the different storage methods in Android?**
- 2. What do you understand about Shared Preferences?**



3. What is Gradle?

4. To write files on the external storage, which permission you will write in AndroidManifest.xml file

5. To check whether the media (external storage) is available, which method of Environment class you will use?

- A) getExternalStorageState()**
- B) getExternalStorage()**
- C) getExternalStateData()**
- D) None of the above.**

Conclusion



PRACTICAL 9

AIM: - Develop an application for working with graphics and animation.

Android Simple Graphics

Here, we are going to see two examples of option menus. First, the simple option menus and second, options menus with images.

Here, we are inflating the menu by calling the `inflate()` method of `MenuInflater` class. To perform event handling on menu items, you need to override `onOptionsItemSelected()` method of `Activity` class.

Canvas and Drawables

Android provides a set of View widgets that provide general functionality for a wide array of user interfaces. You can also extend these widgets to modify the way they look or behave. In addition, you can do your own custom 2D rendering using the various drawing methods contained in the Canvas class or create Drawable objects for things such as textured buttons or frame-by-frame animations.

Hardware Acceleration

Beginning in Android 3.0, you can hardware accelerate the majority of the drawing done by the Canvas APIs to further increase their performance.

OpenGL

Android supports OpenGL ES 1.0 and 2.0, with Android framework APIs as well as natively with the Native Development Kit (NDK). Using the framework APIs is desirable when you want to add a few graphical enhancements to your application that are not supported with the Canvas APIs, or if you desire platform independence and don't demand high performance. There is a performance hit in using the framework APIs compared to the NDK, so for many graphic intensive applications such as games, using the NDK is beneficial (It is important to note though that you can still get adequate performance using the framework APIs. For example, the Google Body app is developed entirely using the framework APIs). OpenGL with the NDK is also useful if you have a lot of native code that you want to port over to Android.



Android Simple Animations

Android provides a large number of classes and interface for the animation development. Most of the classes and interfaces are given in `android.animation` package.

Android Animation enables you to change the object property and behavior at run time. There are various ways to do animation in android.

The *AnimationDrawable* class provides methods to start and end the animation. Even, you can use time based animation.

The Android framework provides two animation systems: property animation (introduced in Android 3.0) and view animation. Both animation systems are viable options, but the property animation system, in general, is the preferred method to use, because it is more flexible and offers more features. In addition to these two systems, you can utilize Drawable animation, which allows you to load drawable resources and display them one frame after another.

Property Animation

Introduced in Android 3.0 (API level 11), the property animation system lets you animate properties of any object, including ones that are not rendered to the screen. The system is extensible and lets you animate properties of custom types as well.

View Animation

View Animation is the older system and can only be used for Views. It is relatively easy to setup and offers enough capabilities to meet many application's needs.

Drawable Animation

Drawable animation involves displaying Drawable resources one after another, like a roll of film. This method of animation is useful if you want to animate things that are easier to represent with Drawable resources, such as a progression of bitmaps.



Activity_main.xml



MainActivity.java





Output

Practice

- 1. What is graphics in Android?**
- 2. What is animation in Android.**



3. List Types of graphics in android.

4. List Types of animation in android.

Conclusion



PRACTICAL 10

AIM: - Create an application that shows your current location.

Android Google Map

Android provides facility to integrate Google map in our application. Google map displays your current location, navigate location direction, search location etc.

We can also customize Google map according to our requirement.

Types of Google Maps

There are four different types of Google maps, as well as an optional to no map at all. Each of them gives different view on map. These maps are as follow:

1. **Normal:** This type of map displays typical road map, natural features like river and some features build by humans.
2. **Hybrid:** This type of map displays satellite photograph data with typical road maps. It also displays road and feature labels.
3. **Satellite:** Satellite type displays satellite photograph data, but doesn't display road and feature labels.
4. **Terrain:** This type displays photographic data. This includes colors, contour lines and labels and perspective shading.
5. **None:** This type displays an empty grid with no tiles loaded.

Android Google Map Displaying Current Location

In the previous tutorial of Android Google Map, we simply displayed the default coordinates (location) set by the *MapsActivity.java* class file.

Now in this tutorial we will display and place marker at the user current location. For doing this we need to generate Google Map API key. The process of generating Google Map API is described in tutorial Android Google Map.

To display the user current location we need to implements some interfaces and there callbacks methods.



Activity_main.xml



MainActivity.java





Output

Practice

- 1. What is geocoding?**

- 2. What does Google Map API do in Android?**



3. What are the tips to reduce the battery usage of your Android app caused due to location usage?

4. For using Google map which permission you will specify in the AndroidManifest.xml?

5. Which element is used to display Google map in your UI.
 - A. View
 - B. Map
 - C. Map View
 - D. None of the above.

Conclusion



PRACTICAL 11

AIM: - Create registration application to store data in database.

Activity_main.xml



MainActivity.java





Output

Practice

- 1. What is the difference between SQL and SQLite?**
- 2. List out the areas where SQLite works well?**



3. List out the advantages of SQLite?

4. List out some features of Android OS.

5. What is the role of Services in Android?

Conclusion