Dr. AMBEDKAR INSTITUTE OF TECHNOLOGY

(An Autonomous Institute, Affiliated to Visvesvaraya Technological University, Belagavi, Accredited by NAAC, with 'A' Grade)
Near JnanaBharathi Campus, Bengaluru – 560056

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Lab Manual

on

"Android Programming"

Subject Code: 18CSL76

Staffs - In – Charge

Ms. Uma K M, Assistant Professor, Dr. AIT
Ms. Lavanya Santhosh, Assistant Professor, Dr. AIT
Ms.Veena A, Assistant Professor, Dr. AIT

2020-21

A. LAORATORY OVERVIEW

Degree:	**BE**	Program:	**CS & E**
Semester:	**6**	Academic Year:	**2020-21**
Laboratory Title:	**Android Programming Laboratory**	Laboratory Code:	**18CSL76**
L-T-P-S:	**0-0-1-0**	Duration of SEE:	**3 Hrs**
CIE Marks:	**50**	SEE Marks:	**50**

B. DESCRIPTION

c

There are no prerequisites for learning Android but I would like to mention that you will be coding in Java and XML. It would be helpful if you could learn just the core Java concepts.

2. BASE COURSE

- **Android Programming Laboratory (18CSL76)**

3. COURSE OUTCOMES

At the end of the course, the student will be able to;

CO1: Create, test and debug Android application by setting up Android development environment.

CO2: Implement adaptive, responsive user interfaces that work across a wide range of devices.

CO3: Infer long running tasks and background work in Android applications.

CO4: Demonstrate methods in storing, sharing and retrieving data in Android applications.

CO5: Infer the role of permissions and security for Android applications.

4. RESOURSES REQUIRED

- **Hardware resources**
  - Desktop PC
  - Microsoft Windows 7/8/10 (32 or 64 bit)
  - 2 GB RAM minimum, 8 GB recommended
  - 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE + 1.5 GB for Android SDK and emulator system image)
  - 1280 x 800 minimum screen resolution
- **Software resources**

All the required tools to develop Android applications are freely available and can be downloaded from the Web. Following is the list of software you will need before you start your Android.

- Java JDK5 or latest version
- Java Runtime Environment (JRE) 6
- Android SDK
- Android Studio
- Eclipse IDE for Java Developers
- Android Development Tool kit (ADT kit) / Eclipse

For testing purpose you require physical device because as per my experience it run smoothly than PC Emulator.

5. RELEVANCE OF THE COURSE:

- To carry out Mini project and Main Project work

6. GENERAL INSTRUCTIONS:

- Implement the program in Android Studio.
- External practical examination.
  - All laboratory experiments are to be included
  - Students are allowed to pick one experiment from the lot.
  - Marks distribution: Procedure + Conduction + Viva: 10 + 30 +10 (50)
  - Change of experiment is allowed only once and marks allotted to the procedure part tobe made zero.

List of Programs

Course objectives:

1) To learn and acquire art of Android programming.
2) To configure initial application, run in emulator.
3) Understand and implement Android's advanced User interface functions, audio video applications
4) Create, modify and query on SQlite database.
5) Present different ways of sharing data through the use of services.

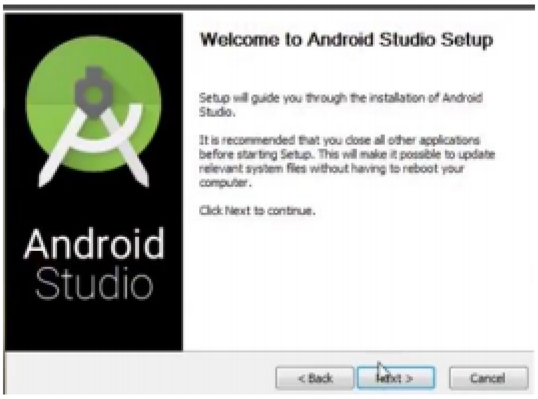
Sl. No.	Programs
1.	i) Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner. The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number. ii) Develop a simple application with one EditText so that the user can write some text in it. Create a button called "Convert Text to Speech" that converts the user input text into voice.
2.	Write a program to create an Activity to read Employee Details (EmpId, Name, Age, Address) from user and store to database and create a menu with menu item (Show Details) on pressing menu details it must go to another activity with employee id search box and search button and display the employee details on the screen.
3.	Write a program to create an activity with a text box and three buttons (save, open and create) open must allow to browse the text file from sdcard and must display the contents of the file on textbox, save button must save the contents of text box to file, create button must allow file user to create a new file and save the entered contents of the textbox.
4.	Write a program to create an activity with two text boxes (date /time and note contents). Create a content provider to store the date and time and note contents to the database. Create another program with a Button (Fetch Today Notes) on press must access the note provider and display the notes stored for today's date.
5.	Write a program to create an activity with two buttons start and stop. On pressing start button the program must start the counter and must keep on counting until stop button is pressed.
6.	Create a program to receive the incoming SMS to the phone and put a notification on screen, on clicking the notification it must display sender number and message content on screen.
7.	Write a program to create a service that will put a notification on the screen every 5 seconds.
8.	Create an .aidl service to do add, subtraction and multiplication and create another application with two buttons to read the inputs and three button add,subtract and multiply to call add, subtract and multiply operation on .aidl service.
9.	Create an activity like a phone dialer with (1,2,3,4,5,6,7,8,9,0,*,#) buttons including call, save and delete buttons. On pressing the call button, it must call the phone number and on pressing the save button it must save the number to the phone contacts.
10.	Create a file of JSON type with values for city_name, Latitude, Longitude, Temperature and Humidity. Develop an application to create an activity with button to parse the JSON file which when clicked should display the data in the textview.

How to Install and Set up Android Studio on Windows?

Step 1: Go to https://developer.android.com **website.**

Step 2: Click on the **Download Android Studio** Button.

Step 3: After the downloading has finished, open the file from downloads and run it. It will prompt the following dialog box.



Welcome to Android Studio Setup

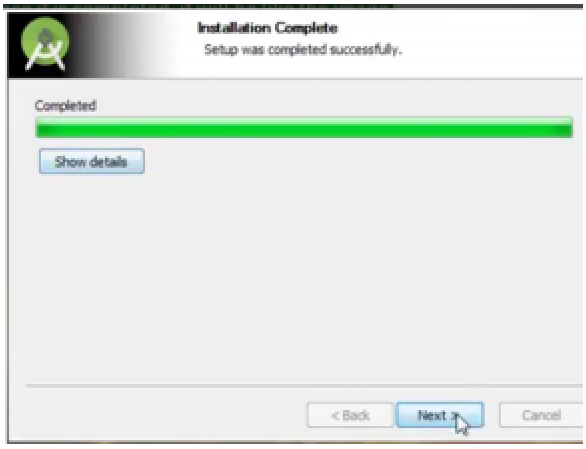
Setup will guide you through the installation of Android Studio.

It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.

Click Next to continue.

Click on next. In the next prompt, it'll ask for a path for installation. Choose a path and hit next.

Step 4: It will start the installation, and once it is completed, it will be like the image shown below.

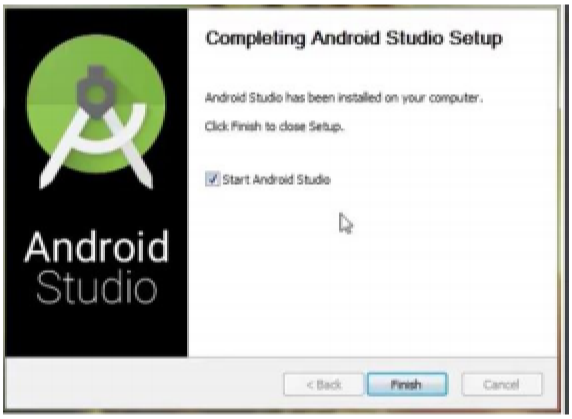


Installation Complete

Setup was completed successfully.

Completed

Click on next.



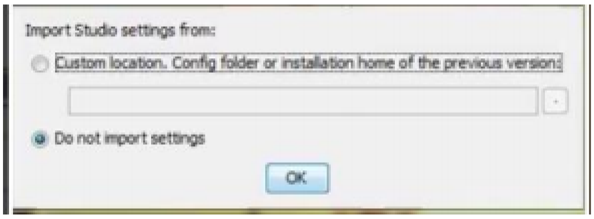
Completing Android Studio Setup

Android Studio has been installed on your computer.

Click Finish to close Setup.

☑ Start Android Studio

Step 5: Once “**Finish**” is clicked, it will ask whether the previous settings need to be imported [if the Android studio had been installed earlier], or not. It is better to choose the ‘Don’t import Settings option’.



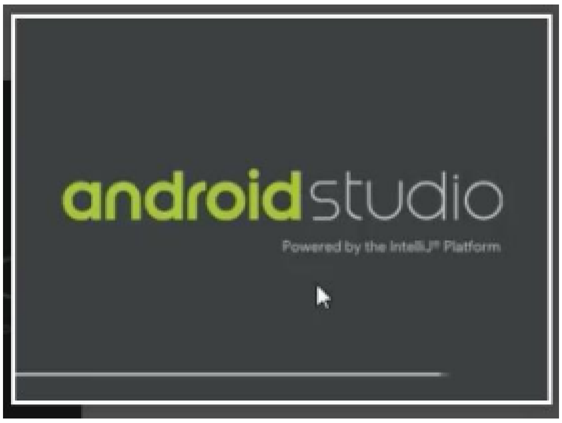
Import Studio settings from:

○ Custom location. Config folder or installation home of the previous version:

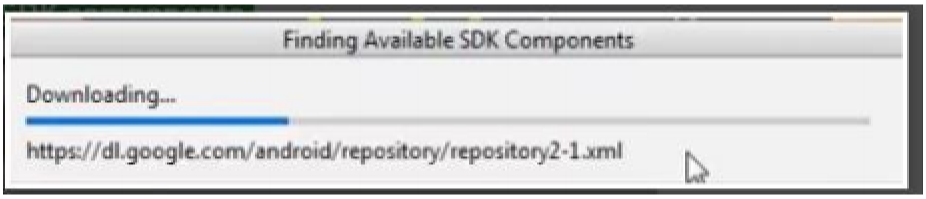
● Do not import settings

Click the **OK** button.

Step 6: This will start the Android Studio.



Meanwhile, it will be finding the available SDK components.

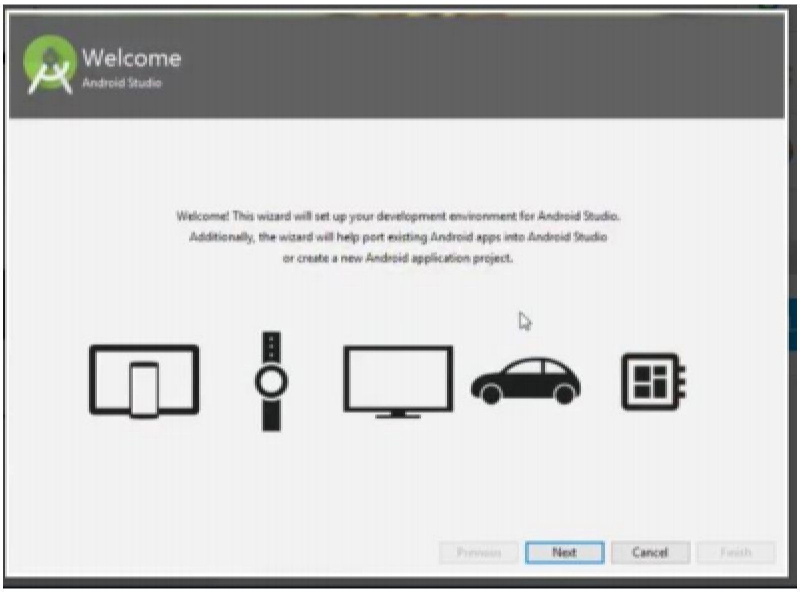


Finding Available SDK Components

Downloading...

https://dl.google.com/android/repository/repository2-1.xml

Step 7: After it has found the SDK components, it will redirect to the Welcome dialog box.

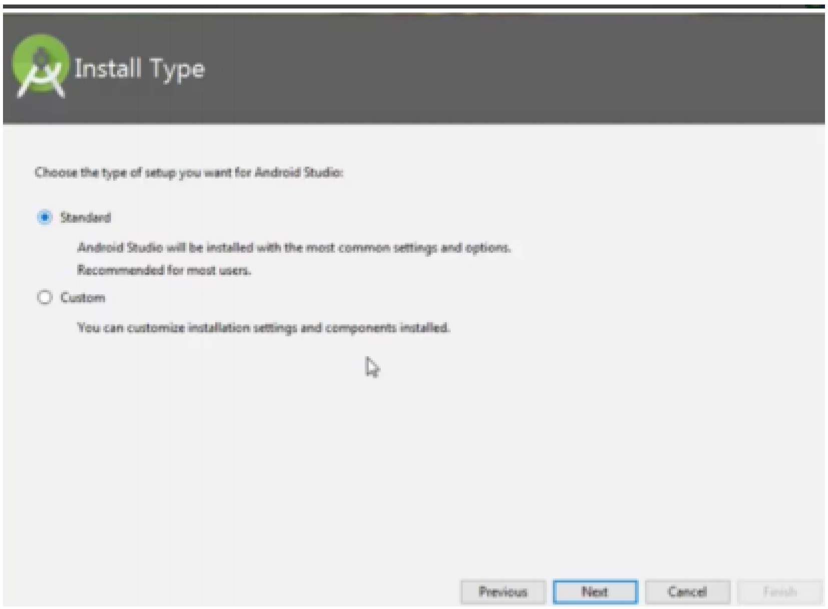


Welcome

Android Studio

Welcome! This wizard will set up your development environment for Android Studio.
Additionally, the wizard will help port existing Android apps into Android Studio
or create a new Android application project.

Click on **Next**.



Install Type

Choose the type of setup you want for Android Studio:

● Standard

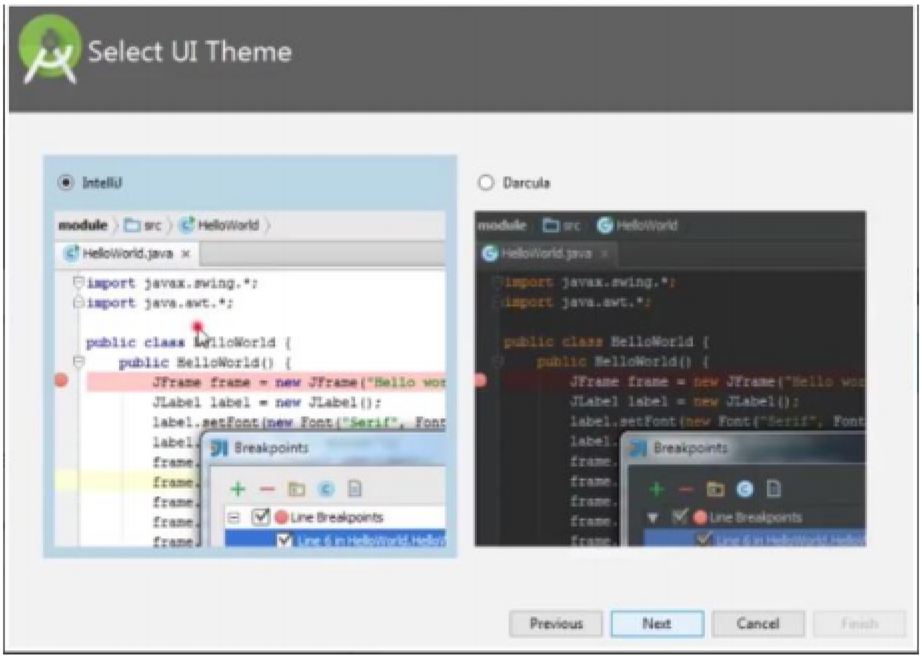
Android Studio will be installed with the most common settings and options.
Recommended for most users.

○ Custom

You can customize installation settings and components installed.

Choose Standard and click on Next. Now choose the theme, whether the **Light** theme or the **Dark** one.

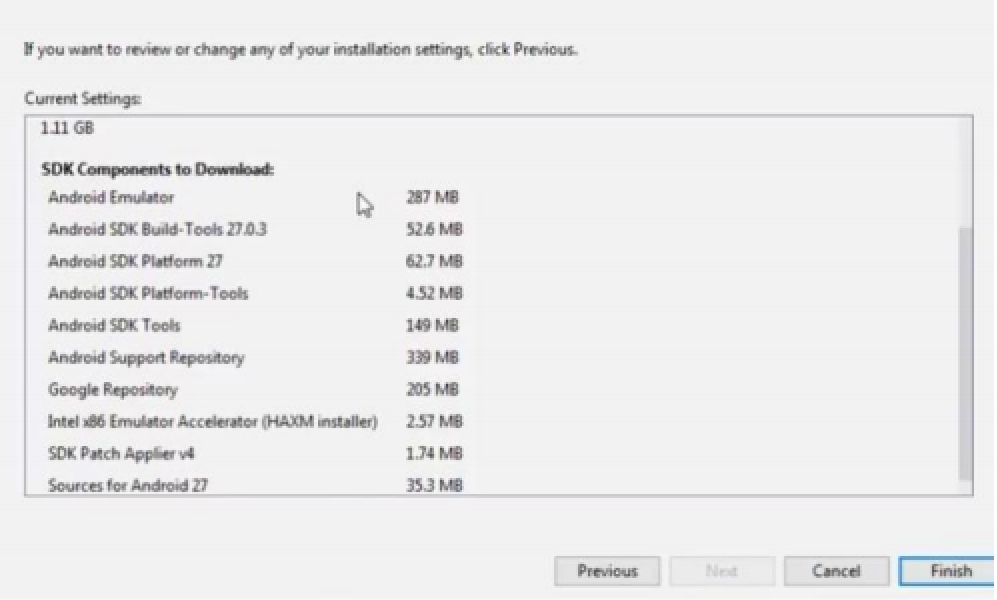
The light one is called the **IntelliJ** theme whereas the dark theme is called **Darcula**. Choose as required.



```
import javax.swing.*;
import java.awt.*;

public class HelloWorld {
    public HelloWorld() {
        JFrame frame = new JFrame("Hello wor
        JLabel label = new JLabel();
        label.setFont(new Font("Serif", Font
        label.
        frame.
        frame.
        frame.
        frame.
        frame.
```

Click on the **Next** button.

Step 8: Now it is time to download the SDK components.



If you want to review or change any of your installation settings, click Previous.

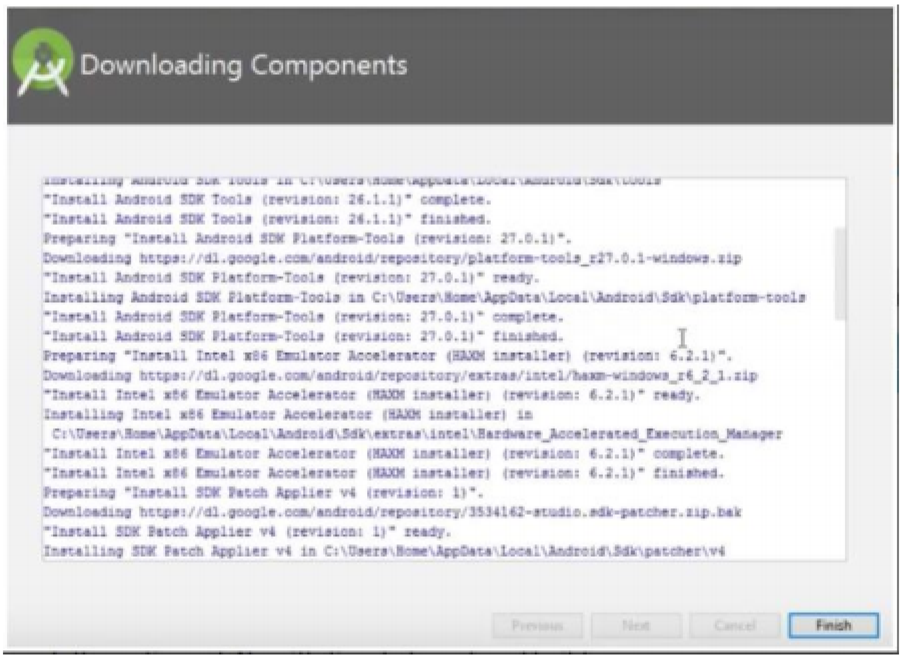
Current Settings:

1.11 GB

SDK Components to Download:

Android Emulator	287 MB
Android SDK Build-Tools 27.0.3	52.6 MB
Android SDK Platform 27	62.7 MB
Android SDK Platform-Tools	4.52 MB
Android SDK Tools	149 MB
Android Support Repository	339 MB
Google Repository	205 MB
Intel x86 Emulator Accelerator (HAXM installer)	2.57 MB
SDK Patch Applier v4	1.74 MB
Sources for Android 27	35.3 MB

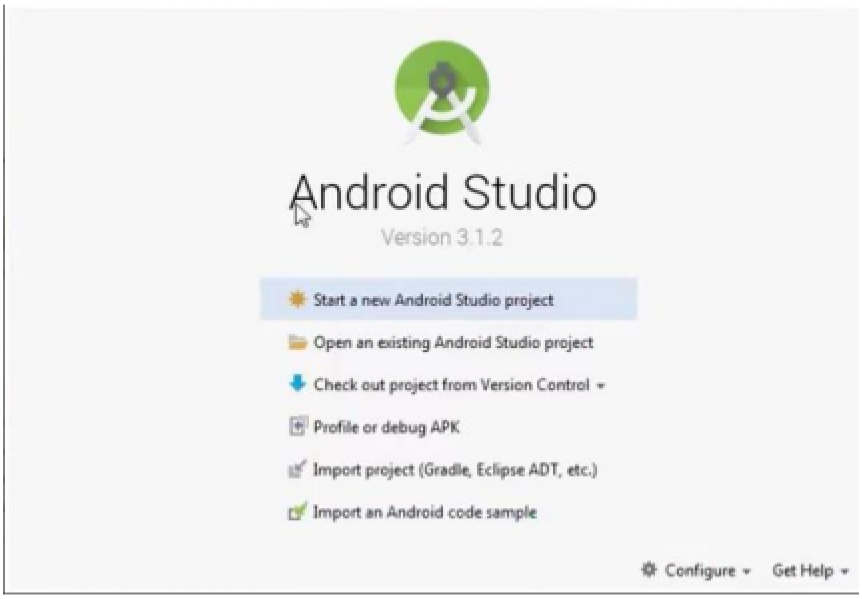
Click on Finish. Components begin to download let it complete.



```
Installing Android SDK Tools in C:\Users\Home\AppData\Local\Android\Sdk\tools
"Install Android SDK Tools (revision: 26.1.1)" complete.
"Install Android SDK Tools (revision: 26.1.1)" finished.
Preparing "Install Android SDK Platform-Tools (revision: 27.0.1)".
Downloading https://dl.google.com/android/repository/platform-tools_r27.0.1-windows.zip
"Install Android SDK Platform-Tools (revision: 27.0.1)" ready.
Installing Android SDK Platform-Tools in C:\Users\Home\AppData\Local\Android\Sdk\platform-tools
"Install Android SDK Platform-Tools (revision: 27.0.1)" complete.
"Install Android SDK Platform-Tools (revision: 27.0.1)" finished.
Preparing "Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)".
Downloading https://dl.google.com/android/repository/extras/intel/haxm-windows_r6_2_1.zip
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" ready.
Installing Intel x86 Emulator Accelerator (HAXM installer) in
 C:\Users\Home\AppData\Local\Android\Sdk\extras\intel\Hardware_Accelerated_Execution_Manager
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" complete.
"Install Intel x86 Emulator Accelerator (HAXM installer) (revision: 6.2.1)" finished.
Preparing "Install SDK Patch Applier v4 (revision: 1)".
Downloading https://dl.google.com/android/repository/3534162-studio.sdk-patcher.zip.bak
"Install SDK Patch Applier v4 (revision: 1)" ready.
Installing SDK Patch Applier v4 in C:\Users\Home\AppData\Local\Android\Sdk\patcher\v4
```

The Android Studio has been successfully configured. Now it's time to launch and build apps. Click on the Finish button to launch it.

Step 9: Click on **Start a new Android Studio project** to build a new app.



Android Studio
Version 3.1.2
Start a new Android Studio project
Open an existing Android Studio project
Check out project from Version Control ▾
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure ▾
Get Help ▾

To run your first android app in Android Studio you may refer to **Running your first Android app.**

Create your First Android Application

Step1: Launch Android Studio.
Step 2: Select File → New → New Project
Step 3: In "Choose your project", select "Phone and Tablet" tab → "Empty Activity" → Next.
Step 4: In "Configure your project" → Set "Name" to "Hello Android" (this will be the "Title" in your phone's application menu) → The "Package name" and "Save Location" will be updated automatically → In "Language", select "Java" → Leave the "Minimum API Level" and the rest to default →Finish.

It could take a few minutes to set up your first app. Watch the "progress bar" at the bottom status bar.

Once the progress bar indicates completion, a hello-world app is created by default.

Setup Emulator (Android Virtual Device (AVD))

To run your Android application under the emulator, you need to first create an Android Virtual Devices (AVD). An AVD models a specific device. You can create AVDs to emulate different android devices (e.g., phone/tablet, android version, screen size, and etc.).

Step 1: In Android studio, select "Tools" → AVD Manager.

Step 2: Click "Create Virtual Device".

Step 3: In "Choose a device definition" → In "Category", choose "Phone" → In "Name", choose "2.7 QVGA" (the smallest device available - you can try a bigger device later) → Next.

Step 4: In "System Image: Recommended" → Select the version with the highest API level → Click "Download" → Next.

Step 5: In "AVD Name", enter "2.7 QVGA API 27" (default) →Finish.

Running the Android Application on Emulator

Step 1: Select the "Run" menu → "Run app" → Under "Available Virtual Devices", select "2.7 QVGA API 27" → OK.

Step 2: It may take a few MINUTES to fire up the app on the emulator. You first see a Google logo → then "Android" → then the "wallpaper" → then the "Hello, world!" message.

Step 3: DO NOT CLOSE THE EMULATOR**, as it really takes a long time to start. You could always re-run the app (or run a new app) on the same emulator. Try re-run the app by selecting "Run" menu → "Run app".**

Program -1

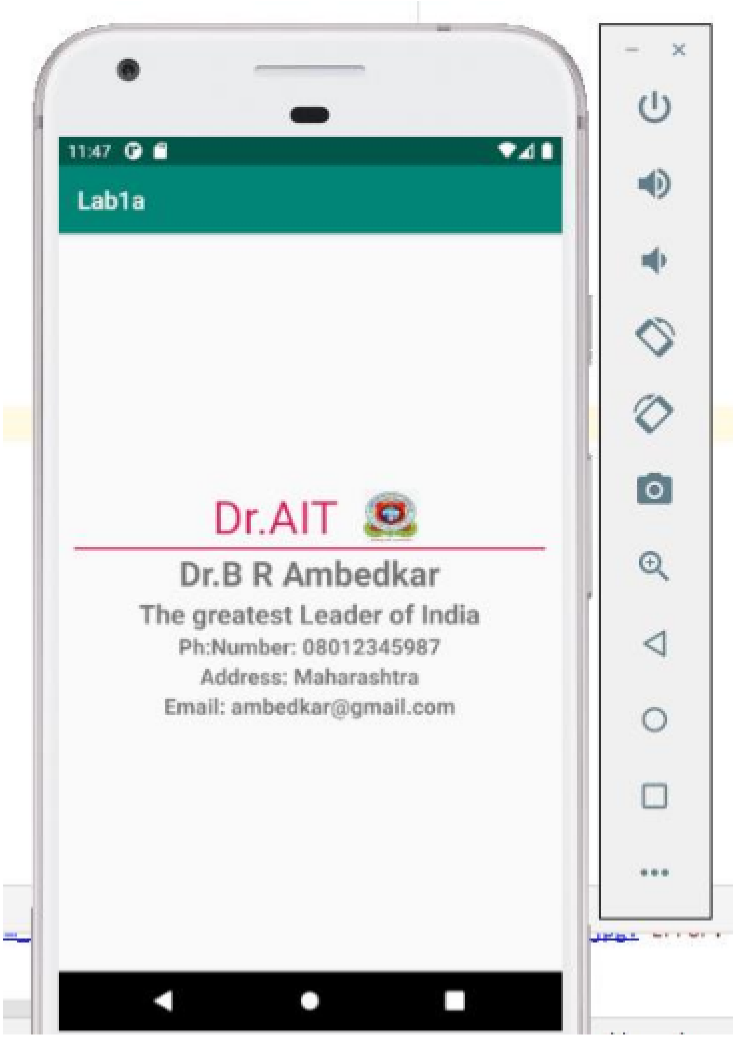
i) Create an application to design a Visiting Card. The Visiting card should have a company logo at the top right corner.The company name should be displayed in Capital letters, aligned to the center. Information like the name of the employee, job title, phone number, address, email, fax and the website address is to be displayed. Insert a horizontal line between the job title and the phone number.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
  -<LinearLayout tools:context=".MainActivity"
    android:gravity="center"
    android:orientation="vertical"
    android:layout_height="match_parent"
    android:layout_width="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:android="http://schemas.android.com/apk/res/android">

  -<LinearLayout
    android:layout_height="wrap_content"
    android:layout_width="match_parent"
    android:layout_gravity="center_horizontal">

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:textSize="38dp"
        android:textColor="@color/colorAccent"
        android:text=" GOOGLE"
        android:id="@+id/textView"/>

  <ImageView
      android:layout_height="wrap_content"
      android:layout_width="wrap_content"
      android:id="@+id/imageView"
      app:srcCompat="@drawable/logo"
      android:layout_marginTop="4dp"/>

  <View
      android:layout_height="4dp"
      android:layout_width="wrap_content"
      android:id="@+id/view"
      android:layout_marginTop="103dp"
      android:background="@color/colorAccent"
      android:layout_marginBottom="498dp"/>

</LinearLayout>
```

```
<TextView
        android:layout_height="30dp"
        android:layout_width="81dp"
        android:text="Dr.B R Ambedkar"
        android:id="@+id/textView3"
        android:layout_marginBottom="16dp"
        android:textStyle="bold"/>

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="The greatest Leader of India"
        android:id="@+id/textView4"
        android:textStyle="bold"/>

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Ph:Number: 2019201923"
        android:id="@+id/textView6"
        android:textStyle="bold"/>

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Address: Maharashtra"
        android:id="@+id/textView7"
        android:textStyle="bold"/>

    <TextView
        android:layout_height="wrap_content"
        android:layout_width="wrap_content"
        android:text="Email: ambedkar@gmail.com"
        android:id="@+id/textView8"
        android:textStyle="bold"/>

</LinearLayout>
```

OUTPUT



Lab1a

Dr.AIT

Dr.B R Ambedkar

The greatest Leader of India

Ph:Number: 08012345987

Address: Maharashtra

Email: ambedkar@gmail.com

Program -1

ii) **Develop a simple application with one EditText so that the user can write some text in it. Create a button called “Convert Text to Speech” that converts the user input text into voice.**

activity_main.xml

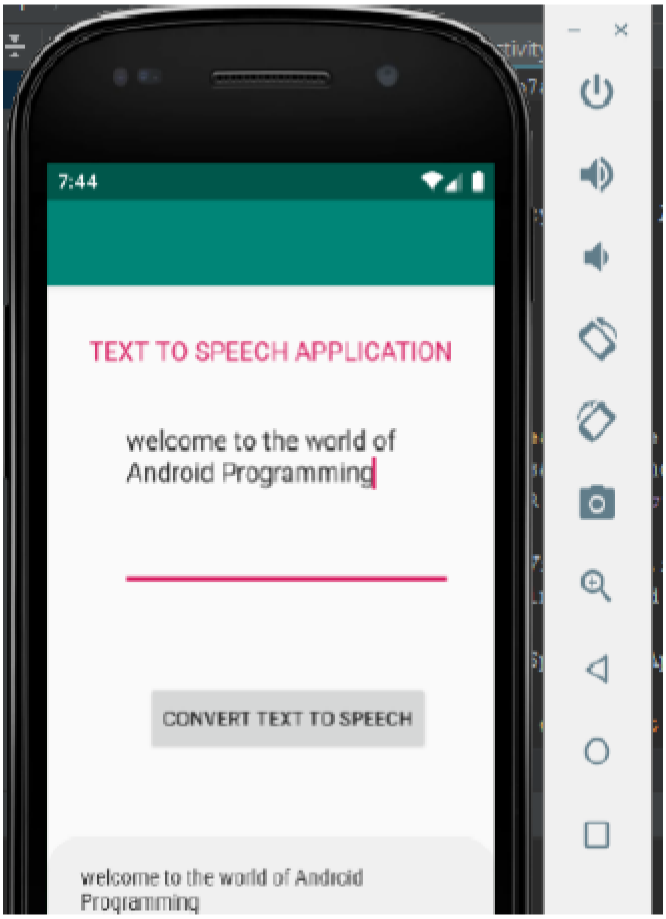
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"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

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="TEXT TO SPEECH APPLICATION"
      android:textSize="18sp"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintEnd_toEndOf="parent"
      app:layout_constraintHorizontal_bias="0.498"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintStart_toStartOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      app:layout_constraintVertical_bias="0.071"     />

   <EditText
      android:id="@+id/txt_input"
      android:layout_width="237dp"
      android:layout_height="177dp"
      android:layout_marginStart="100dp"
      android:layout_marginTop="209dp"
      android:layout_marginEnd="77dp"
      android:layout_marginBottom="437dp"
      android:inputType="textMultiLine”/>

   <Button
      android:id="@+id/btn_txt2spch"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginStart="75dp"
      android:layout_marginTop="399dp"
      android:layout_marginEnd="51dp"
      android:layout_marginBottom="284dp"

      android:text="Convert Text to Speech"
      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

```
package com.example.texttospeech;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.speech.tts.TextToSpeech;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.util.Locale;

public class MainActivity extends AppCompatActivity {

    TextToSpeech t1;
    EditText txtinput;
    Button txttospeech;

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

        txtinput = findViewById(R.id.txt_input);
        txttospeech = findViewById(R.id.btn_txt2spch);

        t1 = new TextToSpeech(getApplicationContext(), new TextToSpeech.OnInitListener() {
            @Override
            public void onInit(int status) {
                if(status != TextToSpeech.ERROR) {
                    t1.setLanguage(Locale.ENGLISH);
                }
            }
        });

        txttospeech.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String tospeak = txtinput.getText().toString();
                Toast.makeText(getBaseContext(),tospeak,Toast.LENGTH_SHORT).show();


               t1.speak(tospeak,TextToSpeech.QUEUE_FLUSH, null);
            }
        });
    }
    public void onPause()
    {
       if(t1 != null)
       {
          t1.stop();
          t1.shutdown();
       }
       super.onPause();    }
}
```

OUTPUT



7:44

TEXT TO SPEECH APPLICATION

welcome to the world of Android Programming

welcome to the world of Android Programming