|  |  |
| --- | --- |
| **Ex.No.1** | **Install and configure Java Development Kit (JDK), android studio and android SDK.** |

**Aim:**

To Install and configure Java Development Kit (JDK), android studio and android SDK

**Procedure:**

1. Installing the Java Development Kit

The Android SDK was developed using the Java programming language. Similarly, Android applications are also developed using Java. As a result, the Java Development Kit (JDK) is the first component that must be installed. Android development requires the installation of either version 6 or 7 of the Standard Edition of the Java Platform Development Kit. Java is provided in both development (JDK) and runtime (JRE) packages. For the purposes of Android development, the JDK must be installed.

2. Downloading the Android Studio Package

Most of the work involved in developing applications for Android will be performed using the Android Studio environment. Android Studio may be downloaded from the following web page:

*http://developer.android.com/sdk/index.html*

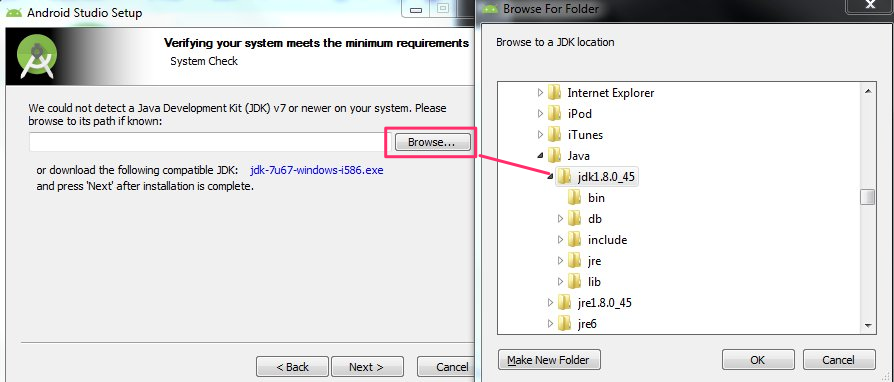
From this page, either click on the download button if it lists the correct platform (for example on a Windows based web browser the button will read “Download Android Studio for Windows”), or select the “Other Download Options” link to manually select the appropriate package for your platform and operating system. On the subsequent screen, accept the terms and conditions to initiate the download.

3. Installing Android Studio

Locate the downloaded Android Studio installation executable file (named *android-studio-bundle-<version>.*exe) in a Windows Explorer window and double click on it to start the installation process, clicking the *Yes* button in the User Account Control dialog if it appears.

Once the Android Studio setup wizard appears, work through the various screens to configure the installation to meet your requirements in terms of the file system location into which Android Studio should be installed and whether or not it should be made available to other users of the system. Although there are no strict rules on where Android Studio should be installed on the system, the remainder of this book will assume that the installation was performed into a sub-folder of the user’s home directory named *android-studio*. Once the options have been configured, click on the *Install* button to begin the installation process.

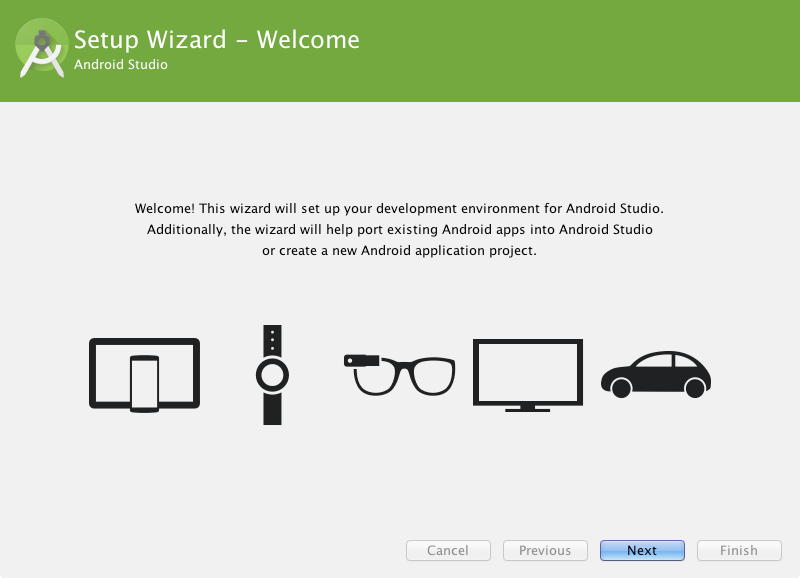




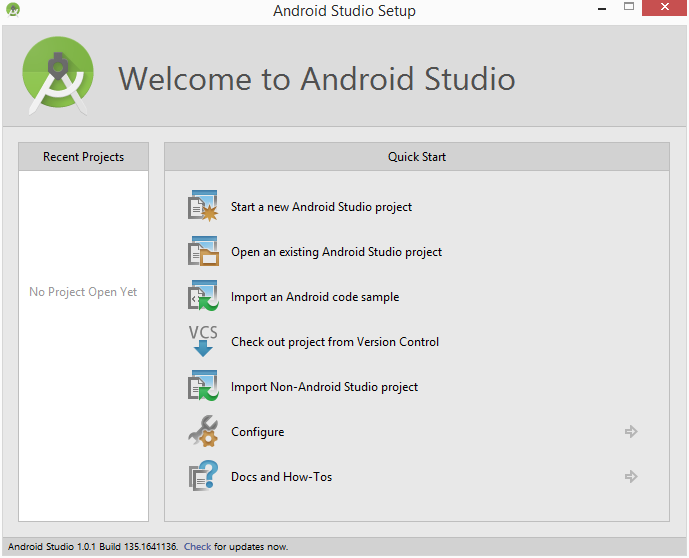
4. The Android Studio Setup Wizard

The first time that Android Studio is launched after being installed, a dialog will appear providing the option to import settings from a previous Android Studio version. If you have settings from a previous version and would like to import them into the latest installation, select the appropriate option and location. Alternatively, indicate that you do not need to import any previous settings and click on the OK button to proceed.

After Android Studio has finished loading, the setup wizard will appear as shown



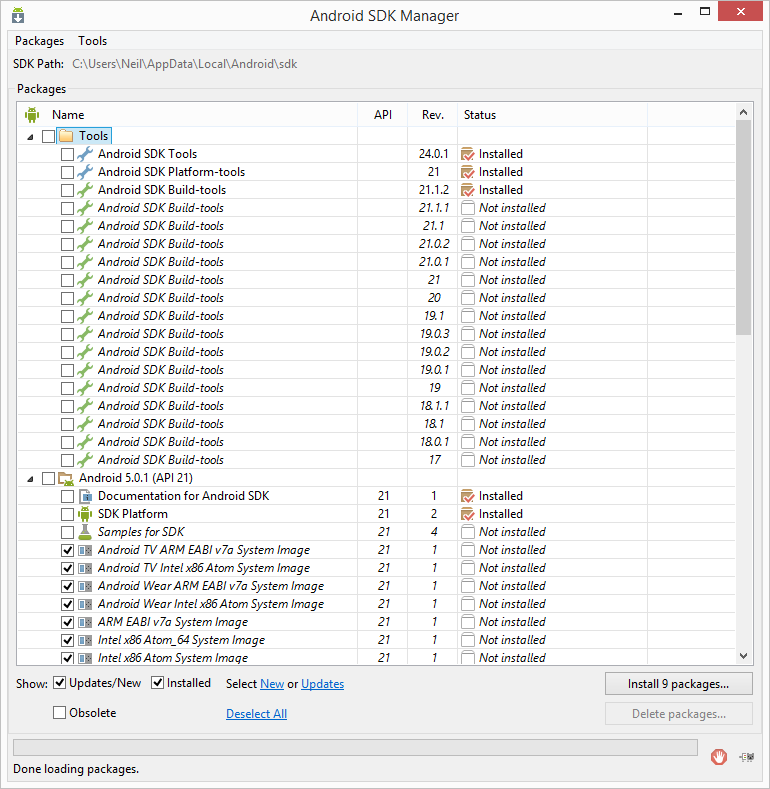
Click on the Next button, choose the Standard installation option and click on Next once again. On the license agreement screen, select and accept each of the licenses listed before clicking on Finish to complete the setup process. The Welcome to Android Studio screen should then appear:



5. Installing the Latest Android SDK Packages

The steps performed so far have installed Java, the Android Studio IDE and the current set of default Android SDK packages. Before proceeding, it is worth taking some time to verify which packages are installed and to install any missing packages.

This task can be performed using the *Android SDK Manager*, which may be launched from within the Android Studio tool by selecting the *Configure -> SDK Manager* option from within the Android Studio welcome dialog. Once invoked, the SDK Manager tool will appear as illustrated in Figure



Within the Android SDK Manager, make sure that the following packages are listed as *Installed* in the Status column:

 Tools > Android SDK Tools

 Tools > Android SDK Platform-tools

 Tools > Android SDK Build-tools

 SDK Platform (most recent version) > SDK Platform

 SDK Platform (most recent version) > ARM EABI v7a System Image

 Extras > Android Support Repository

 Extras > Android Support Library

 Extras > Google Repository

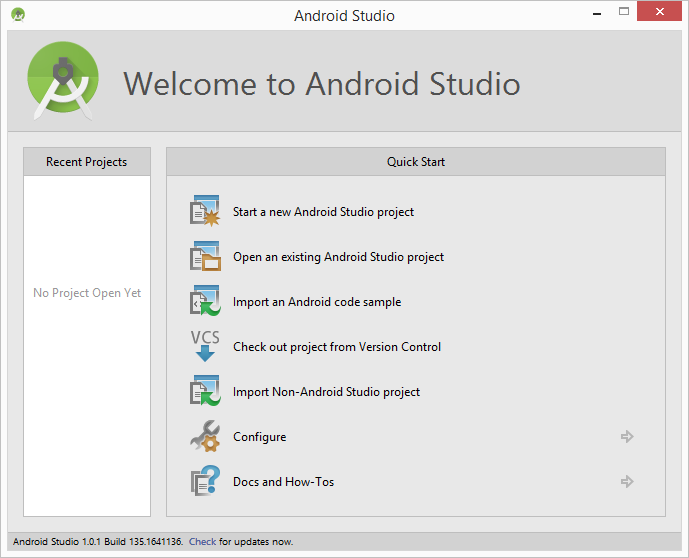
 Extras > Google USB Driver (Required on Windows systems only)

 Extras > Intel x86 Emulator Accelerator (HAXM installer)

In the event that any of the above packages are listed as *Not Installed*, simply select the checkboxes next to those packages and click on the *Install packages* button to initiate the installation process. In the resulting dialog, accept the license agreements before clicking on the *Install* button. The SDK Manager will then begin to download and install the designated packages. As the installation proceeds, a progress bar will appear at the bottom of the manager window indicating the status of the installation.

Once the installation is complete, review the package list and make sure that the selected packages are now listed as *Installed*in the *Status* column. If any are listed as *Not installed,* make sure they are selected and click on the *Install packages…* button again.

6. Creating a New Android Project

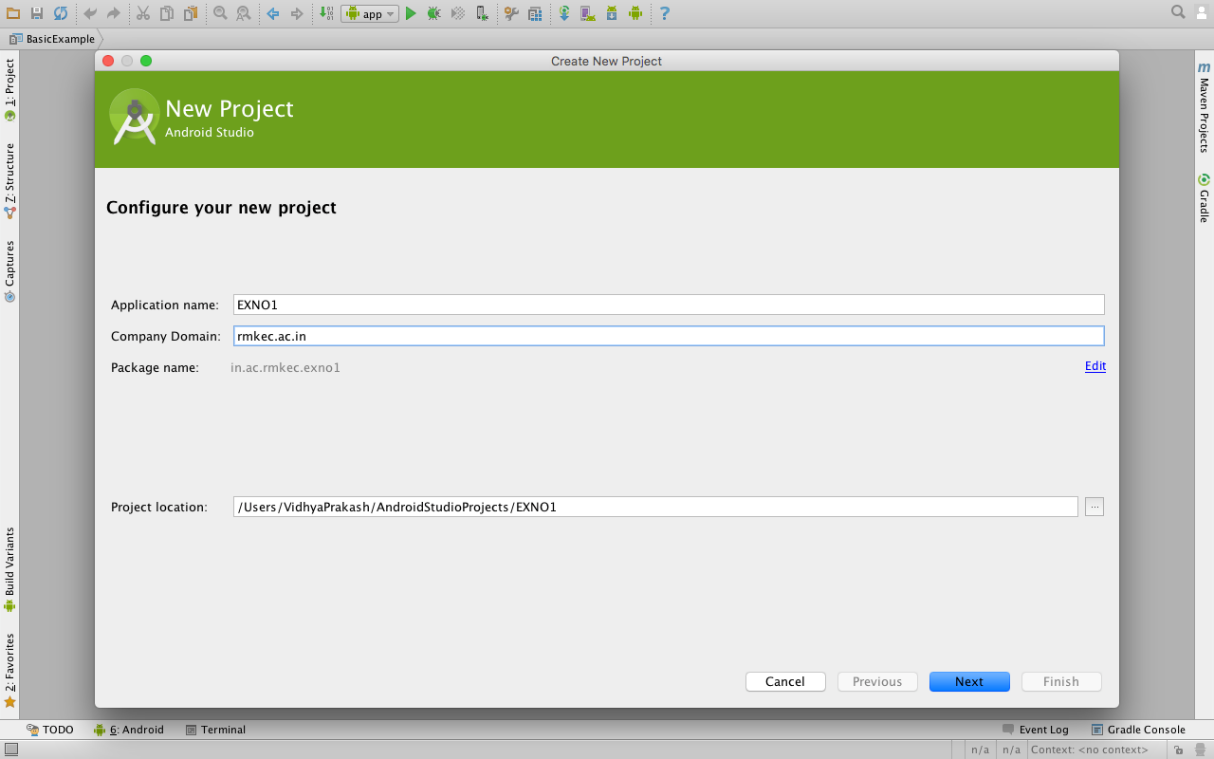
The first step in the application development process is to create a new project within the Android Studio environment. Begin, therefore, by launching Android Studio so that the “Welcome to Android Studio” screen appears as illustrated in Figure

Once this window appears, Android Studio is ready for a new project to be created. To create the new project, simply click on the *Start a new Android Studio project* option to display the first screen of the *New Project* wizard as shown in Figure

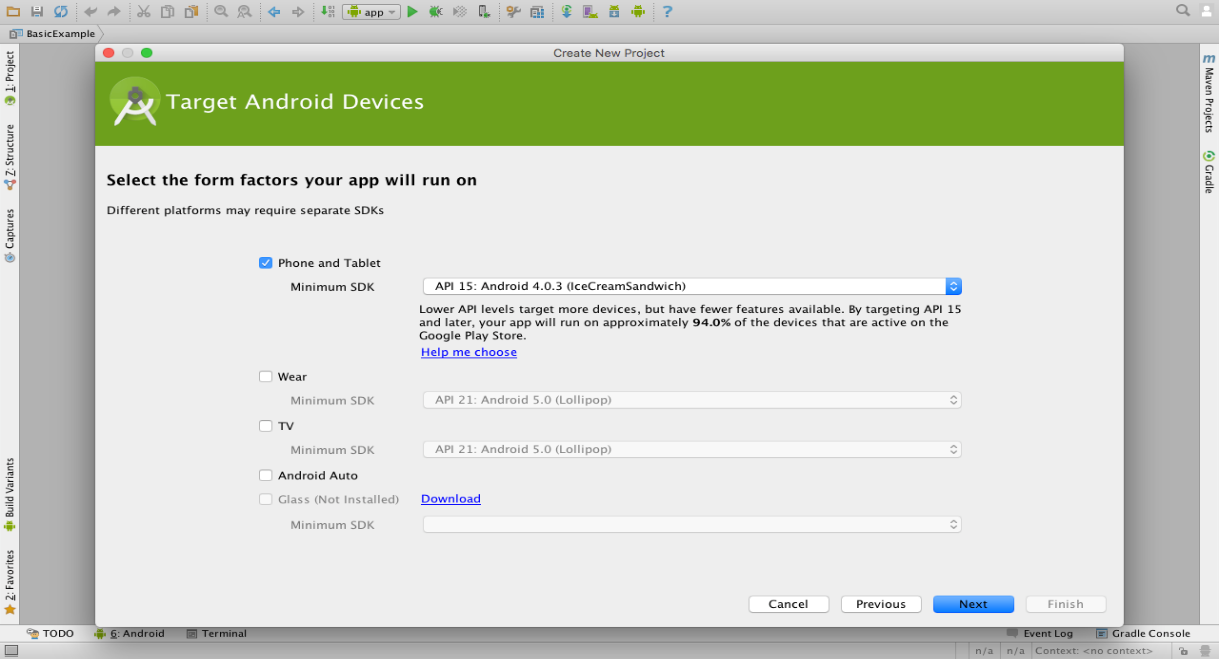
7. Defining the Project and SDK Settings

In the *New Project* window, set the *Application name* field to *EXNO1*. The application name is the name by which the application will be referenced and identified within Android Studio and is also the name that will be used when the completed application goes on sale in the Google Play store.

The *Package Name* is used to uniquely identify the application within the Android application ecosystem. It should be based on the reversed URL of your domain name followed by the name of the application. For example, if your domain is *rmkec.ac.in*, and the application has been named *EXNO1*, then the package name might be specified as follows:



Next select Android package SDK we need to build



**Result:**

The Java Development Kit (JDK), android studio and android SDK are installed.

|  |  |
| --- | --- |
| **Ex.No.2** | **Develop an application that uses GUI components, Font and Colours** |

**Aim:**

To develop an application that uses GUI components,Fonts and Colors.

**Procedure:**

Step 1: File New 🡪 Android Project Application

Step 2: Give the Project Name 🡪 Next

Step 3: Go to Res Folder and Select Layout Click the Main.xml

Step 4: Design the Graphical Layout

Step 5: Write code the in MainActivity.java

Step 6: Run the code using AVD Emulator

**ActivityMain.Java**

packagecom.example.text;

importandroid.app.Activity;

importandroid.graphics.Color;

importandroid.os.Bundle;

importandroid.view.Menu;

importandroid.view.View;

importandroid.view.View.OnClickListener;

importandroid.widget.Button;

importandroid.widget.TextView;

public class MainActivity extends Activity {

Button b1,b2;

TextView t1;

float font;

inti=1;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

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

t1=(TextView)findViewById(R.id.textView1);

b1.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View arg0) {

// TODO Auto-generated method stub

t1.setTextSize(font);

font=font+4;

if(font==40)

font=20;

}

});

b2=(Button)findViewById(R.id.button2);

b2.setOnClickListener(new OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

switch (i) {

case 1:

t1.setTextColor(Color.parseColor("#0000FF"));

break;

case 2:

t1.setTextColor(Color.parseColor("#00FF00"));

break;

case 3:

t1.setTextColor(Color.parseColor("#FF0000"));

break;

default:

t1.setTextColor(Color.parseColor("#800000"));

break;

}

i++;

if(i==5)

i=1;

}

});

}

@Override

publicbooleanonCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.main, menu);

return true;

}

}

**activity\_main.xml**

<RelativeLayoutxmlns: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:paddingBottom=*"@dimen/activity\_vertical\_margin"*

android:paddingLeft=*"@dimen/activity\_horizontal\_margin"*

android:paddingRight=*"@dimen/activity\_horizontal\_margin"*

android:paddingTop=*"@dimen/activity\_vertical\_margin"*

tools:context=*".MainActivity"*>

<TextView

android:id=*"@+id/textView1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignParentTop=*"true"*

android:layout\_centerHorizontal=*"true"*

android:layout\_marginTop=*"140dp"*

android:text=*"@string/hello\_world"* />

<Button

android:id=*"@+id/button1"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_below=*"@+id/textView1"*

android:layout\_centerHorizontal=*"true"*

android:layout\_marginTop=*"78dp"*

android:text=*"Change Font Size"*

android:textSize=*"23sp"*/>

<Button

android:id=*"@+id/button2"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_alignLeft=*"@+id/button1"*

android:layout\_below=*"@+id/button1"*

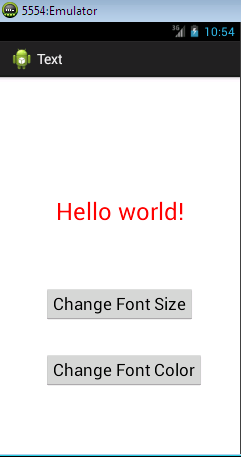
android:layout\_marginTop=*"40dp"*

android:text=*"Change Font Color"*

android:textSize=*"23sp"* />

</RelativeLayout>

**Output**



**Result:**

Thus the program to develop an application that uses GUI components, Fonts and Colors is executed successfully.