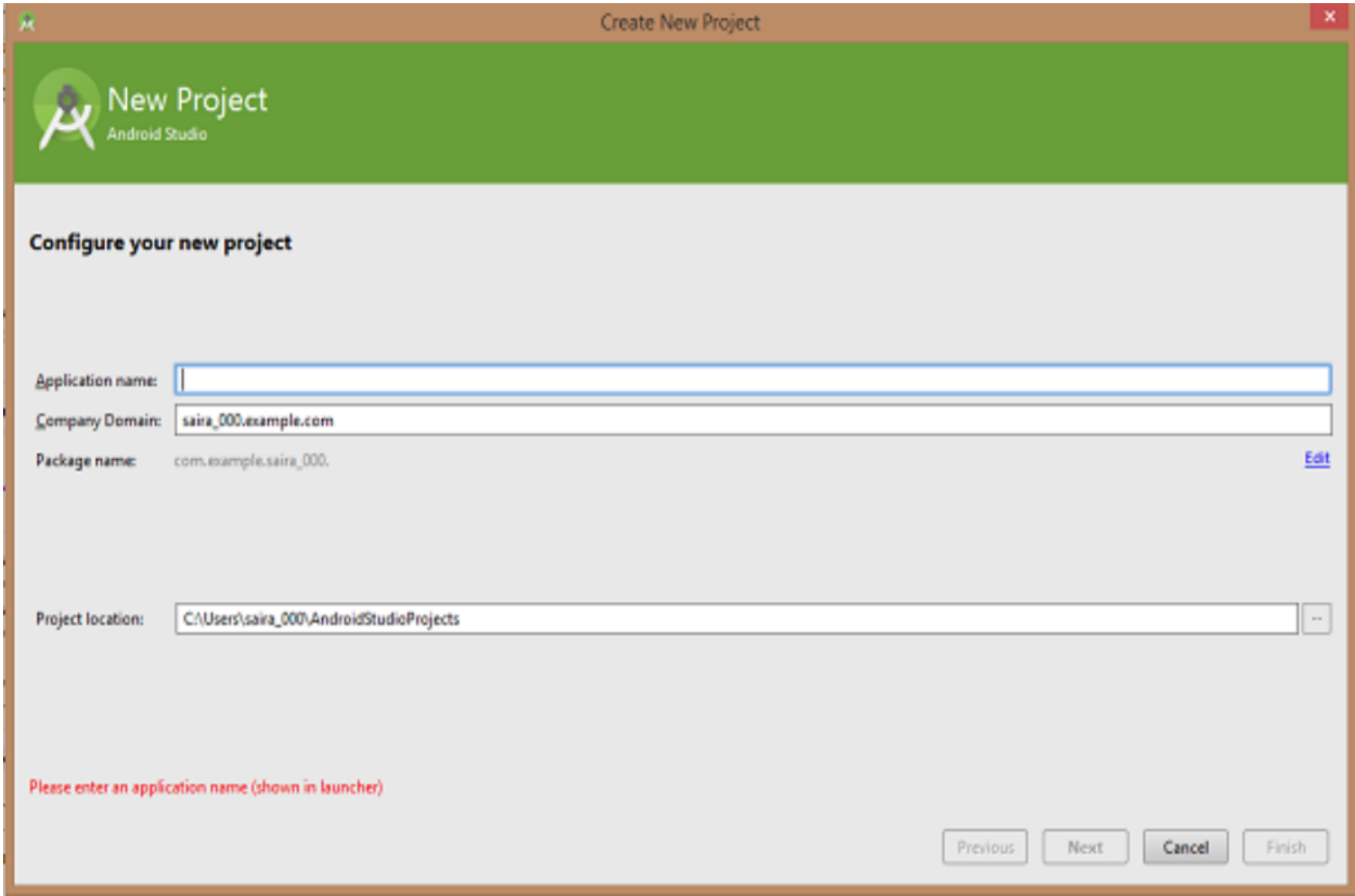
SMART DEVICE PROGRAMMING

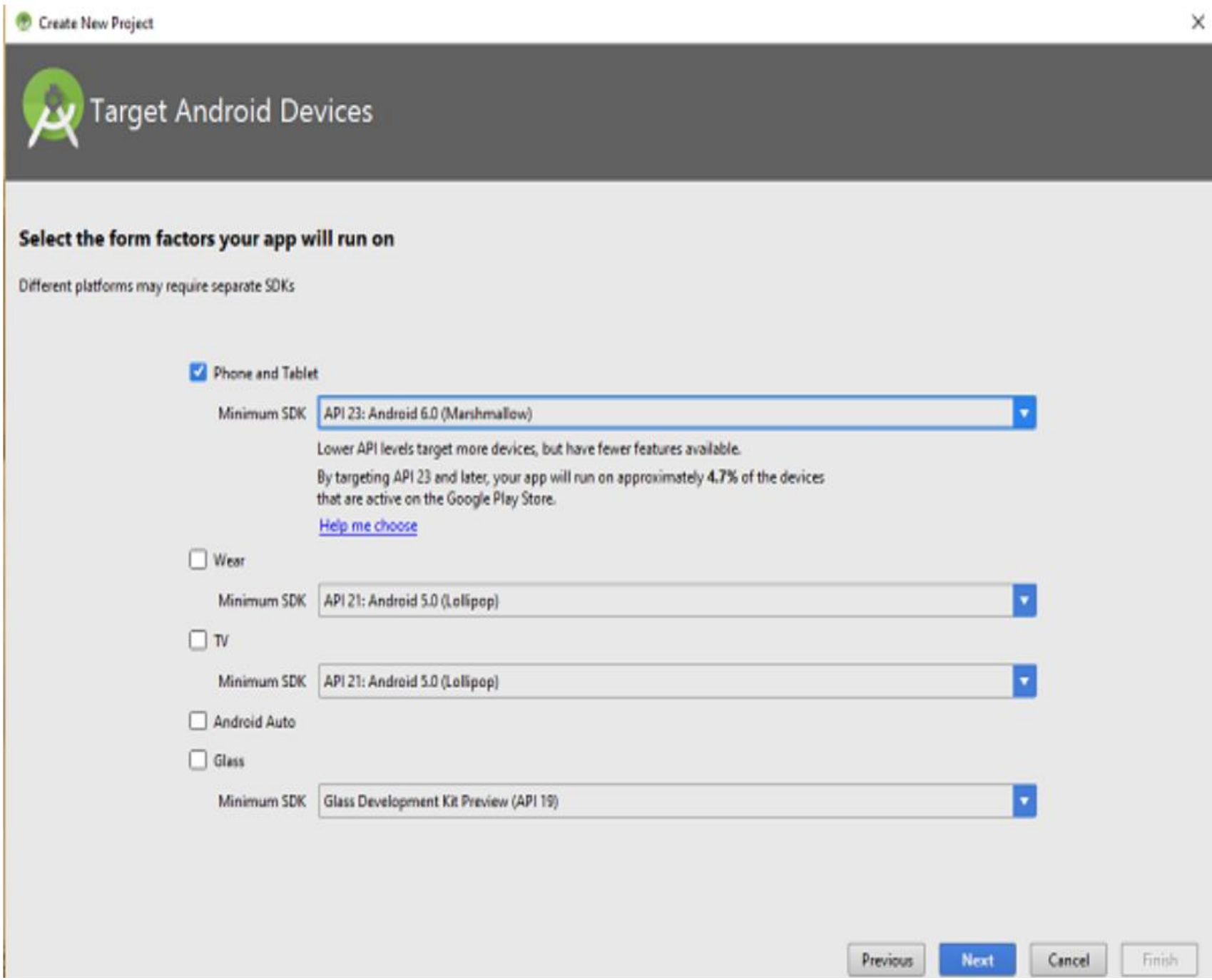
6133

A SIMPLE ANDROID APPLICATION WHICH WILL PRINT "HELLO WORLD!"

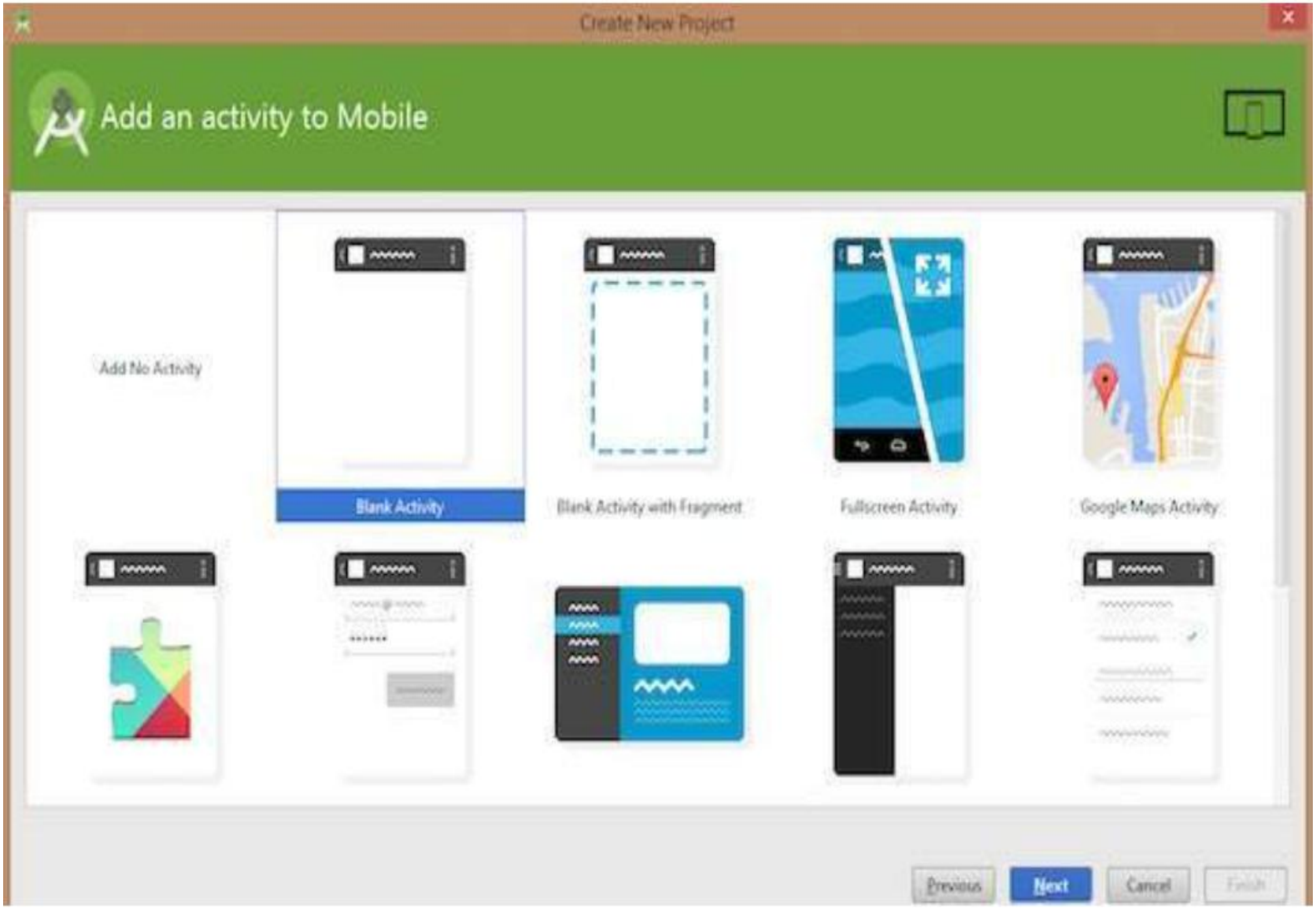
- ✓ Click on Android studio

Start your application development by calling start a new android studio project. In a new installation frame should ask Application name, package information and location of the project.

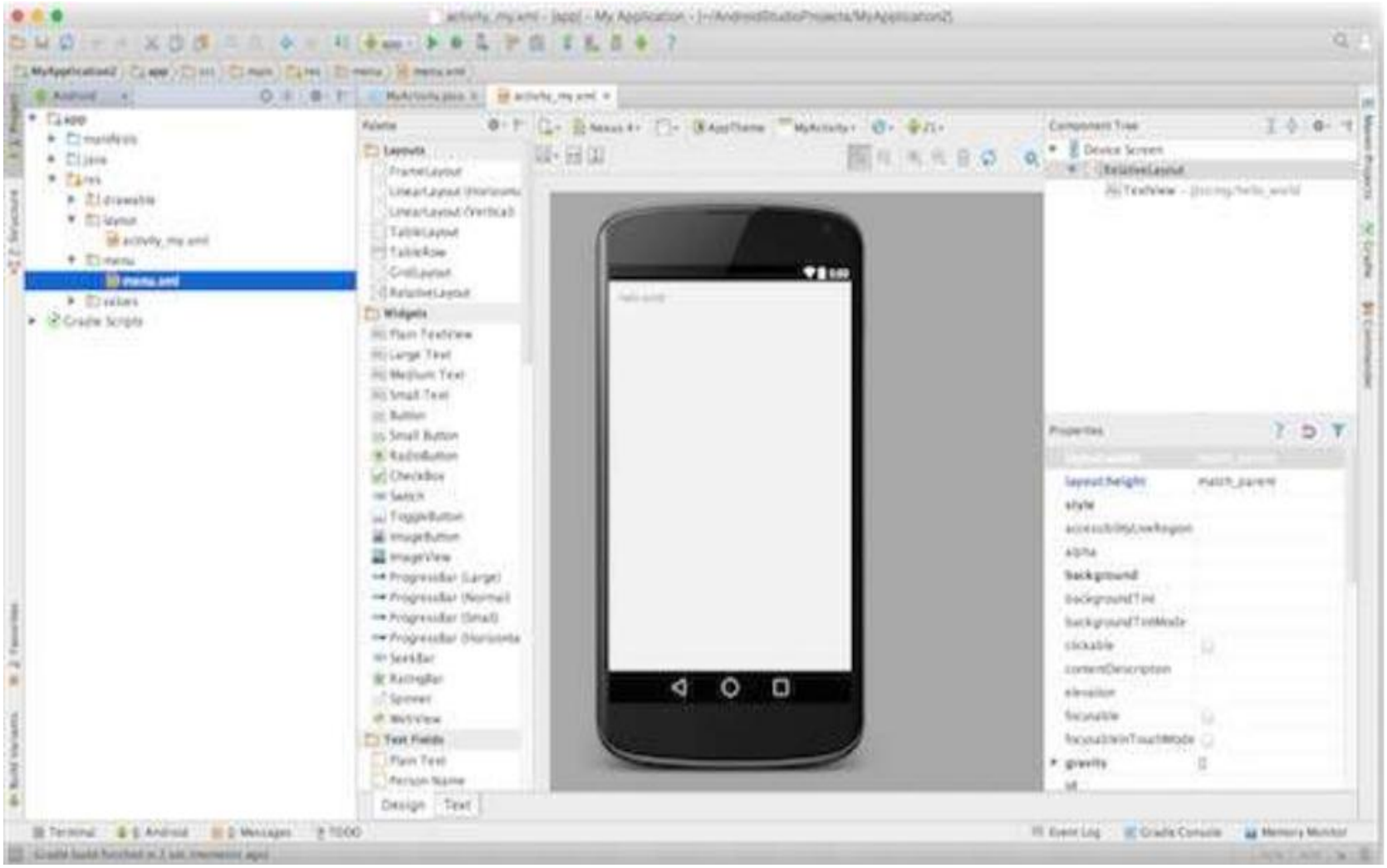



Create New Project
Target Android Devices
Select the form factors your app will run on
Different platforms may require separate SDKs
Phone and Tablet
Minimum SDK
API 23: Android 6.0 (Marshmallow)
Lower API levels target more devices, but have fewer features available.
By targeting API 23 and later, your app will run on approximately 4.7% of the devices that are active on the Google Play Store.
Help me choose
Wear
Minimum SDK
API 21: Android 5.0 (Lollipop)
TV
Minimum SDK
API 21: Android 5.0 (Lollipop)
Android Auto
Glass
Minimum SDK
Glass Development Kit Preview (API 19)
Previous
Next
Cancel
Finish

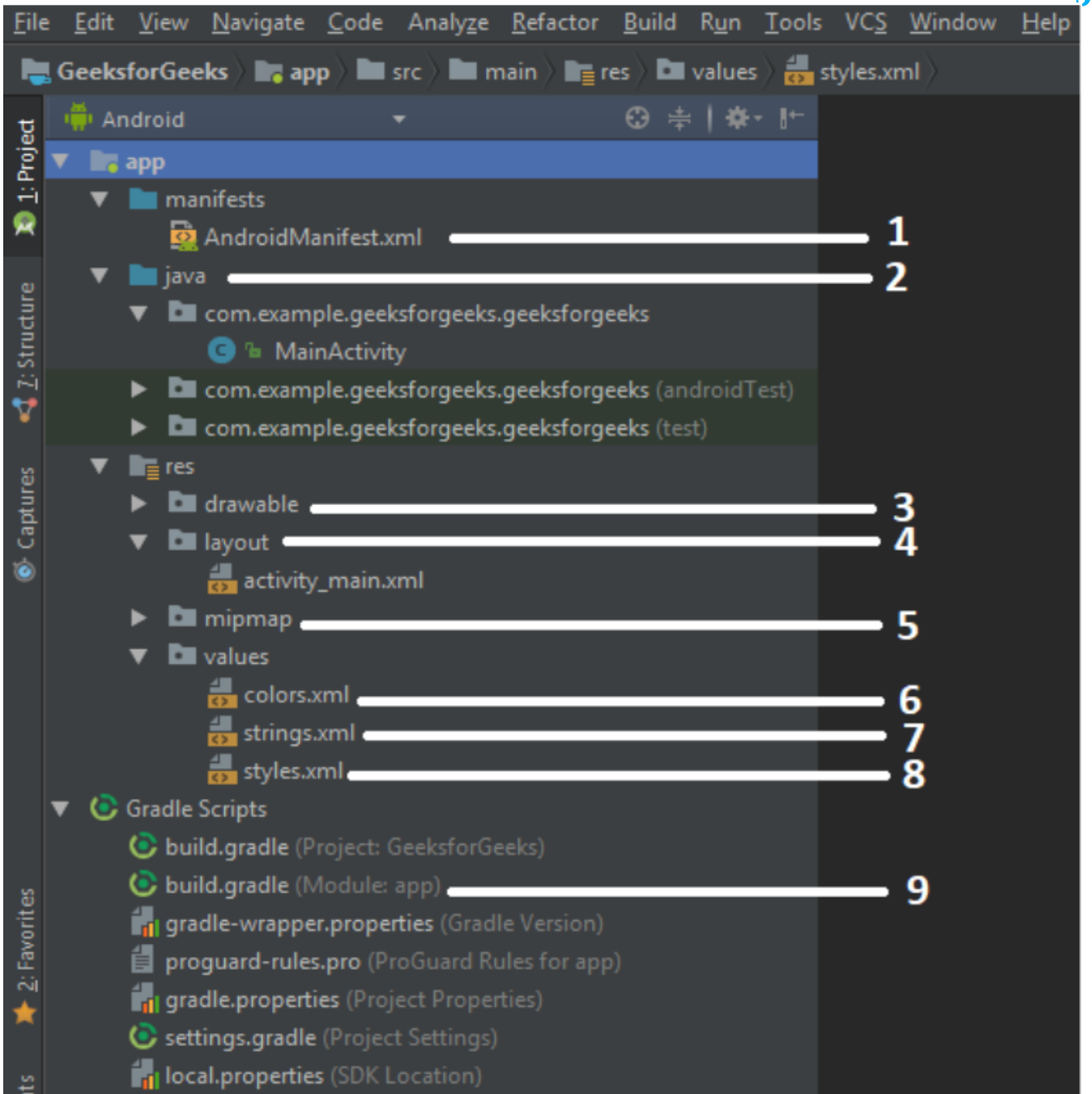
The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications.



At the final stage it going to be open development tool to write the application code



Directories And Files In The Android Project



Android app module

- ✓ Provides a container for app’s source code , resource files and app level settings.
- ✓ Major sub divisions are
  - Manifest
  - Java
  - Res

AndroidManifest.xml

- ✓ Every project in Android include a manifest file , **AndroidManifest.xml** stored in the root directory of its project hierarchy.
- ✓ It defines the structure and metadata of our application , its components and requirements.
- ✓ This file includes nodes for each of the Activities , Services , Content providers and Broadcast receivers that make the application and using intent filters and permissions , determines how they co-ordinate with each other and other applications.

Java

- ✓ The Java folder contains java source code files.
- ✓ These files are used as a controller for controlled UI (Layout File)
- ✓ It gets the data from the layout file and after processing that data output will be shown in the UI layout.
- ✓ It works on the backend of an android application.

Res

- ✓ Resource folder is the most important folder because it contain all the non-code sources like images , XML layout , UI strings for our android application.
- ✓ Major subdivisions are
  - Drawable
  - Layout
  - Mipmaps
  - Values

Drawable

- ✓ A drawable folder contains resource type files (Something that can be drawn).
- ✓ Drawable may take a variety of file like mipmap (PNG , JPEG) , Nine patch , Vector (XML) , Shape , Layers , States , Levels and Scales.

Layout

- ✓ Defines the visual structure for a user interface, such as the UI for an android application.
- ✓ This folder stores Layout files that are written in XML language.

Mipmap

- ✓ Contains Launcher.xml files to define icons which are used to show on the home screen.

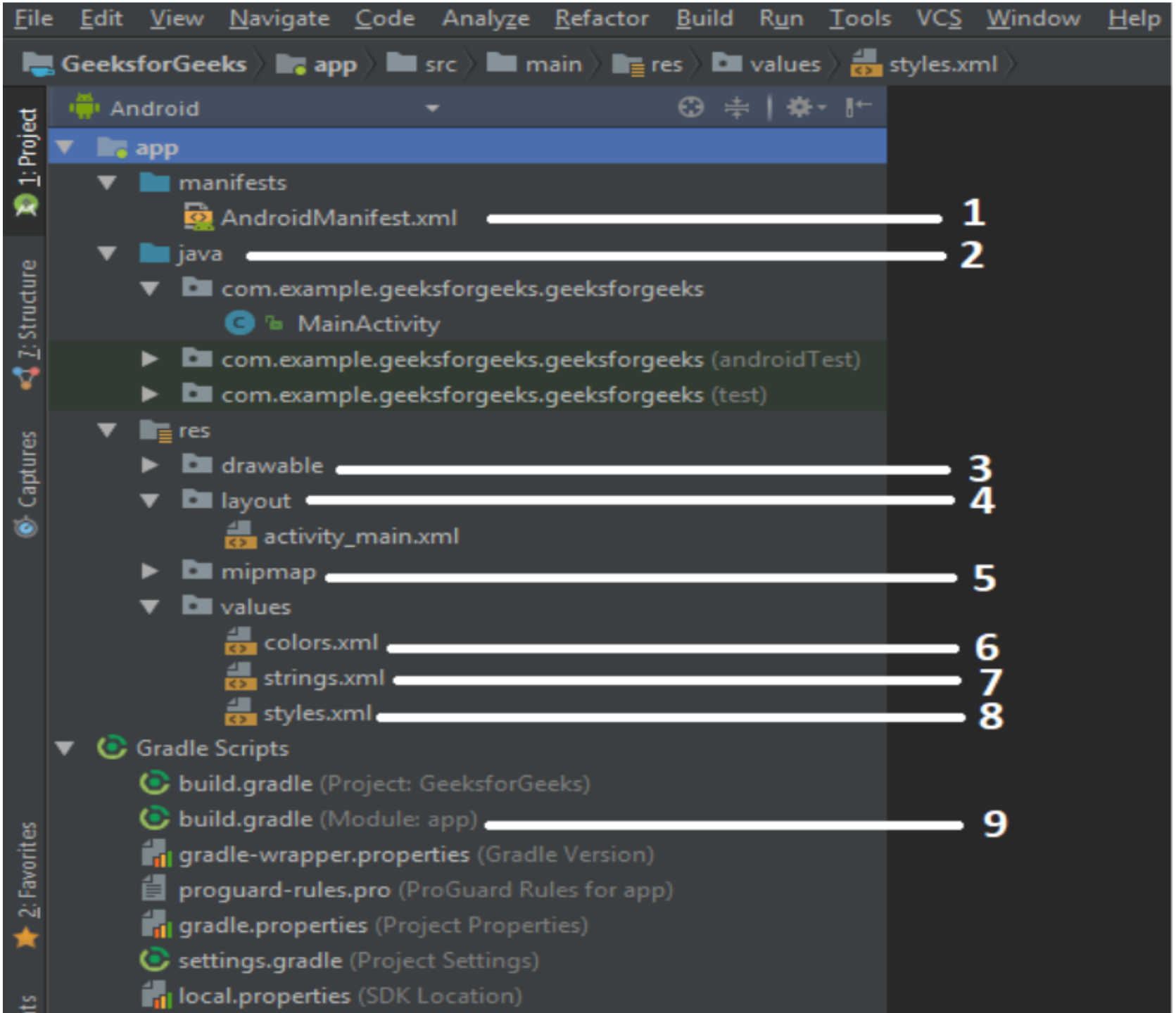
Values

- ✓ Value folder contains a number of XML files like strings , dimens , colors and styles definitions.
  - **colors.xml :**
  - ✓ Contains colour resources of the android application.
  - ✓ Different color values are identified by a unique name that can be used in the android application program.
  - **strings.xml:**
  - ✓ Contains string resources of the android application.
  - ✓ The different string value is identified by a unique name that can be used in the android application program.
  - ✓ This file also stores string array by using XML language.

- **styles.xml**
  - ✓ The styles.xml file contains resources of the theme style in the android application.
  - ✓ This file is written in XML language.

Gradle script

- ✓ Gradle is a build system which is used to automate building , testing , deployment etc.
- ✓ Every android project needs a gradle for generating an apk from the .java and .xml files in the project.
- ✓ A gradle takes all the source files (java and xml) and apply appropriate tools, eg; converts the java files into dex files and compress all of them into a single file known as apk that is actually used.


File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
GeeksforGeeks app src main res values styles.xml
Android
app
manifests
AndroidManifest.xml
1
java
2
com.example.geeksforgeeks.geeksforgeeks
MainActivity
com.example.geeksforgeeks.geeksforgeeks (androidTest)
com.example.geeksforgeeks.geeksforgeeks (test)
res
drawable
3
layout
4
activity_main.xml
mipmap
5
values
colors.xml
6
strings.xml
7
styles.xml
8
Gradle Scripts
build.gradle (Project: GeeksforGeeks)
build.gradle (Module: app)
9
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
1: Project
7: Structure
Captures
2: Favorites

Manifest.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.tutorialspoint7.myapplication">

   <application
      android:allowBackup="true"
      android:icon="@mipmap/ic_launcher"
      android:label="@string/app_name"
      android:supportsRtl="true"
      android:theme="@style/AppTheme">

      <activity android:name=".MainActivity">
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />
            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>
   </application>
</manifest>
```

MainActivity.java

```
package com.example.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

Activity_main.xml

```
<RelativeLayout xmlns: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" >

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_centerVertical="true"
      android:padding="@dimen/padding_medium"
      android:text="@string/hello_world"
      tools:context=".MainActivity" />

</RelativeLayout>
```


5:00
Hello world!