INTRODUCTION

MOBILE COMPUTING
&
APPLICATION DEVELOPMENT

Subject Code: **3360704**

Teaching & Examination Scheme

Teaching Scheme (In Hours)			Total Credits (L+T+P)	Examination Scheme				
				Theory Marks		Practical Marks		Total Marks
L	T	P	C	ESE	PA	ESE	PA	200
3	0	4	7	70	30	40	60	

Teaching Hours & Marks (Theory)

Unit No.	Unit Title	Teaching Hours	Distribution of Theory Marks			
			R Level	U Level	A Level	Total Marks
I	Introduction to Mobile Computing	14	10	10	2	22
II	Introduction to Android	6	2	6	2	10
III	Android Activities and GUI Design concepts.	8	2	4	8	14
IV	Advanced UI Programming	6	2	2	6	10
V	Toast, Menu, Dialog, List and Adapters	8	2	6	6	14
	Total	**42**	**18**	**28**	**24**	**70**

UNIT - II

INTRODUCTION TO ANDROID

Compiled by M.R. Thakkar

2.1 Overview of Android

- Android is a mobile operating system that is based on a modified version of Linux. It was originally developed by Andy Rubin who has been credited as the father of the android platform. His company Android Inc., was acquired by the Google in 2005 and took over its development work as well as its development team.

- Google wanted Android to be open and free, hence the Android code was released as the open source.

- Simply, Android is a combination of:
  - A free, open-source operating system for mobile devices
  - An open-source development platform for creating mobile applications

2.2 Open Handset Alliance

- The Open Handset Alliance (OHA) was formed in November 2007, which is a group of more than 50 technology companies including handset manufacturers, chip manufacturers, software developers and service providers that introduced Android, an open source mobile phone operating system.

- Some of the well known mobile technology companies like Motorola, HTC, T-Mobile, and Qualcomm are members of OHA, however, several major companies and manufacturers are absent from the alliance, including Nokia, Symbian, Apple, RIM(Research In Motion's), Microsoft, Verizon and Cingular.

2.3 What does Android run On

- The first Android mobile handset, the T-Mobile G1, was released in the United States in October 2008 was developed by handset manufacturer HTC with service provided by T-mobile. By the end of 2009 over 20 Android-compatible handsets had been launched or announced in more than 26 countries on 32 different carrier networks.

- Today, Android devices come in all shapes and sizes. The Android OS powers the following types of the devices:
  - Smartphones
  - Tablets
  - E-reader devices
  - Netbooks
  - MP4 players
  - Internet TVs

2.4 Why Android for mobile apps development?

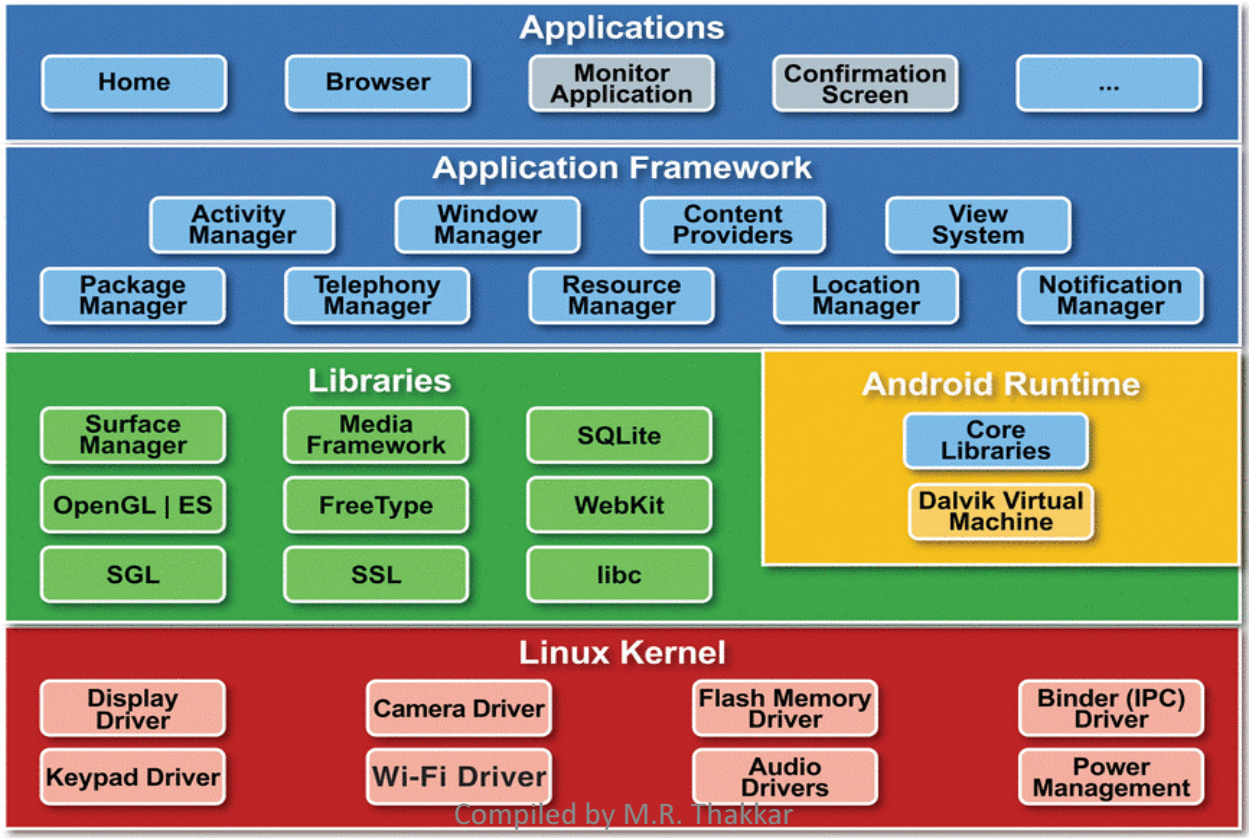
- Android has many innovative features as:
  - Free and Open Source
  - Familiar and inexpensive Development Tools:
  - Enabling Development of powerful applications
  - No Costly Obstacles to Publication
  - A “Free Market” for Applications

2.5 Environment setup for Android apps Development

- To get started, you'll need to download and install the following:
  - Java Development Kit (JDK) 5 or above
    www.oracle.com/technetwork/java/javase/downloads/index.htm
  - The Android SDK
    http://developer.android.com/sdk/index.html
  - Eclipse (Optional)
    www.eclipse.org/downloads/

2.6 Android: Framework, SDK

- **2.6.1 Android Framework**
- In order to understand how Android works, we required to learn about the framework of the Android, following figure shows the various layers that make the framework of the Android operating system.
- The Android OS framework is consist of five sections arranged in four layers:



2.6 Android: Framework, SDK

- ## 2.6.1 Android Framework

Linux kernel: The Core services including hardware drivers, process and memory management, security, network, and power management are handled by a Linux kernel. The kernel also provides an abstraction layer between the hardware and the remainder of the stack.

Libraries: Android libraries run on top of the kernel, Android include various C/C++ core libraries such as libc and SSL, as well as:

- A media library for playback of audio and video media
- A surface manager to provide display management
- Graphics libraries that include SGL and OpenGL for 2D and 3D graphics
- SQLite for native database support
- SSL and WebKit for integrated web browser and Internet security

Android run time: Including the core libraries and the Dalvik virtual machine, the Android run time is the engine that powers your applications and, along with the libraries, forms the basis for the application framework.

2.6 Android: Framework, SDK

- ## 2.6.1 Android Framework

Application framework The application framework provides the classes used to create Android applications. It also provides a generic abstraction for hardware access and manages the user interface and application resources.

Application layer All applications, both native and third-party, are built on the application layer by means of the same API libraries. The application layer runs within the Android run time, using the classes and services made available from the application framework.

2.6 Android: Framework, SDK

- **2.6.2 Android SDK**

  In order to build an android application, it is required to download Android Software Development Kit (SDK). The Android SDK includes everything you need to start developing, testing, and debugging Android applications. The Android SDK consist of:

  - The Android APIs
  - Development tools
  - The Android Virtual Device Manager and Emulator
  - Full documentation
  - Sample code
  - Online support (developer.android.com)

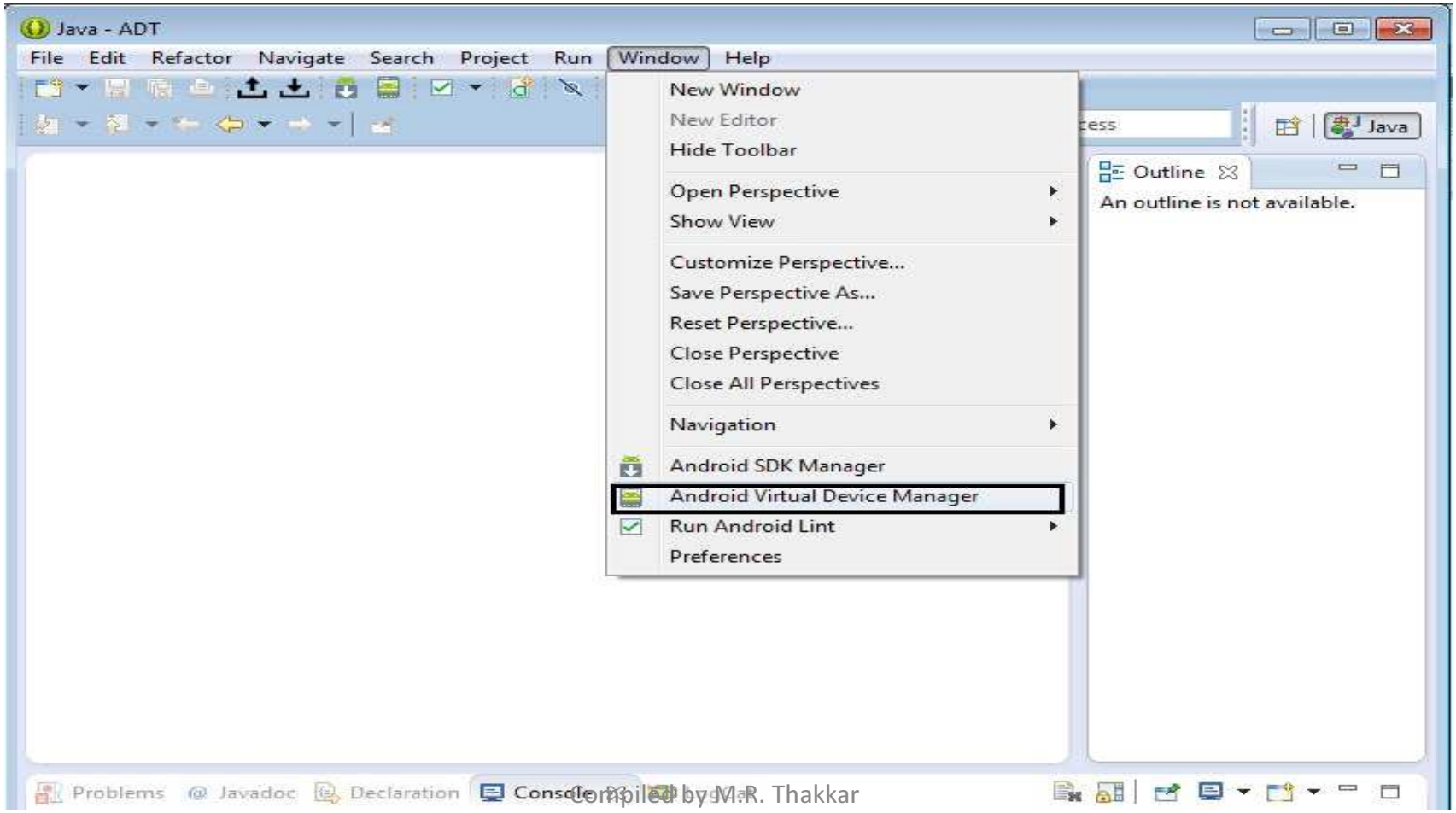
2.7 What is an Emulator / Android AVD

- The Android Emulator is a fully interactive Android device emulator featuring several alternative skins. The emulator runs within an Android Virtual Device that simulates the device hardware configuration.

- Using the emulator you can see how your applications will look and behave on a real Android device.

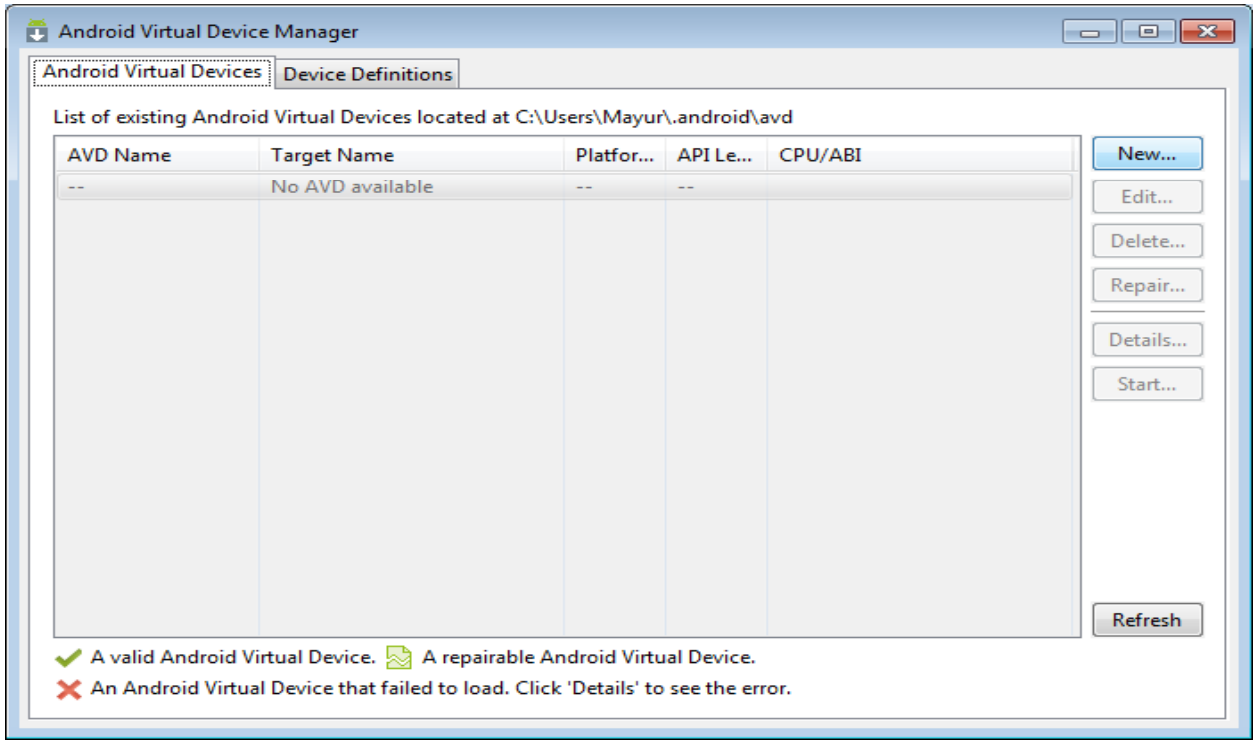
2.8 Android Emulation – Creation and set up

- **STEP 1:** Select **Window ➪ Android Virtual Device Manager.**



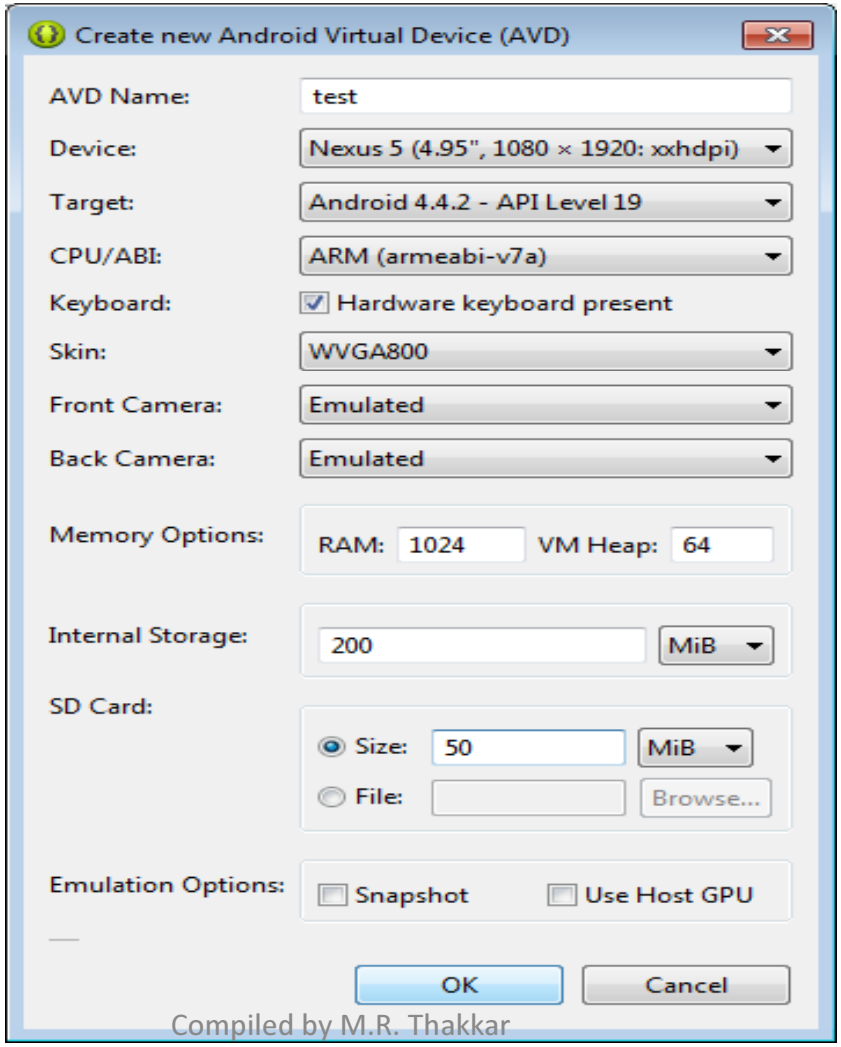
2.8 Android Emulation – Creation and set up

- **STEP 2:** In the Android Virtual Device Manager dialog , click the **New...** button to create a new AVD.



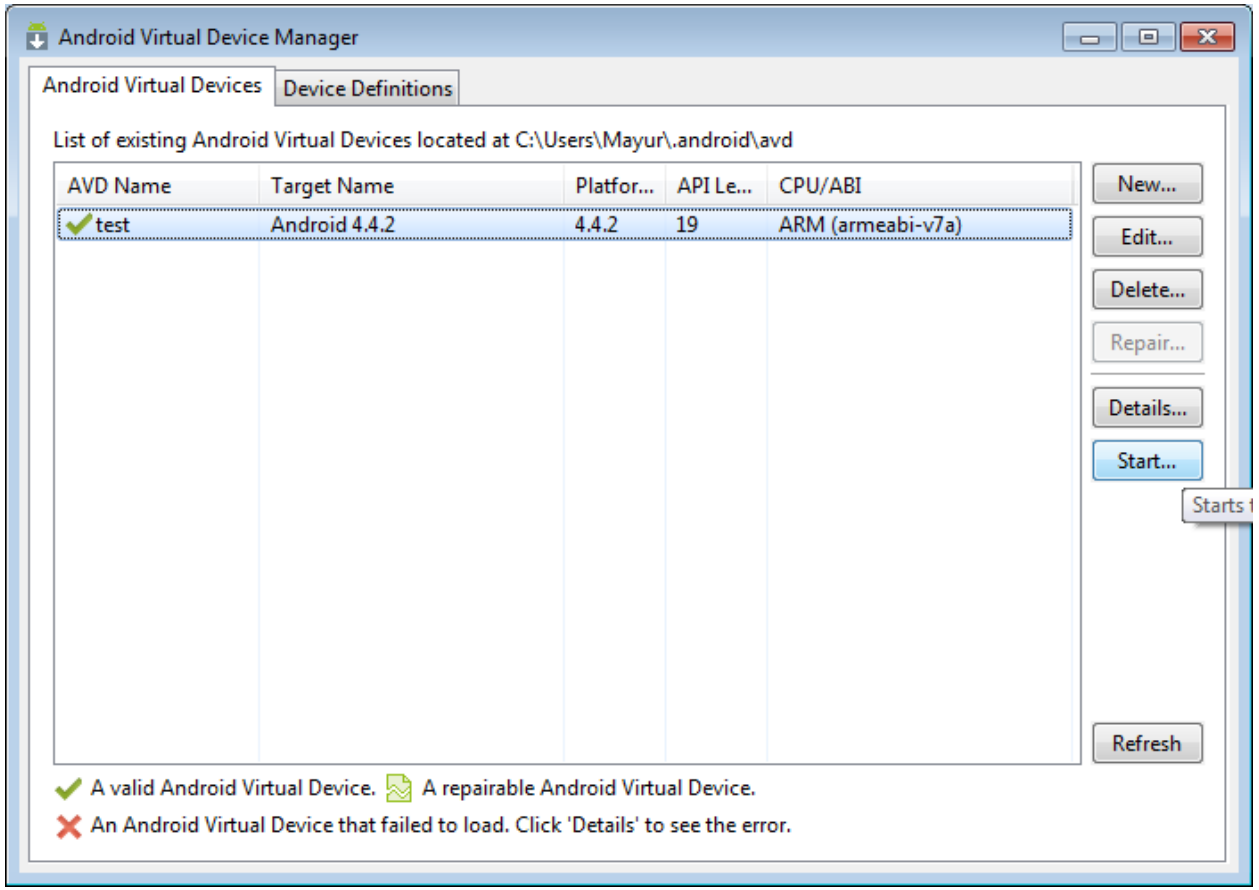
2.8 Android Emulation – Creation and set up

- **STEP 3:** In the Create new Android Virtual Device (AVD) dialog, enter the items as shown in Figure and Click the **OK** button when you are done.



2.8 Android Emulation – Creation and set up

- **STEP 4:** Once your AVD has been created, it is time to test it. Select the AVD that you want to test and click the **Start…** button.



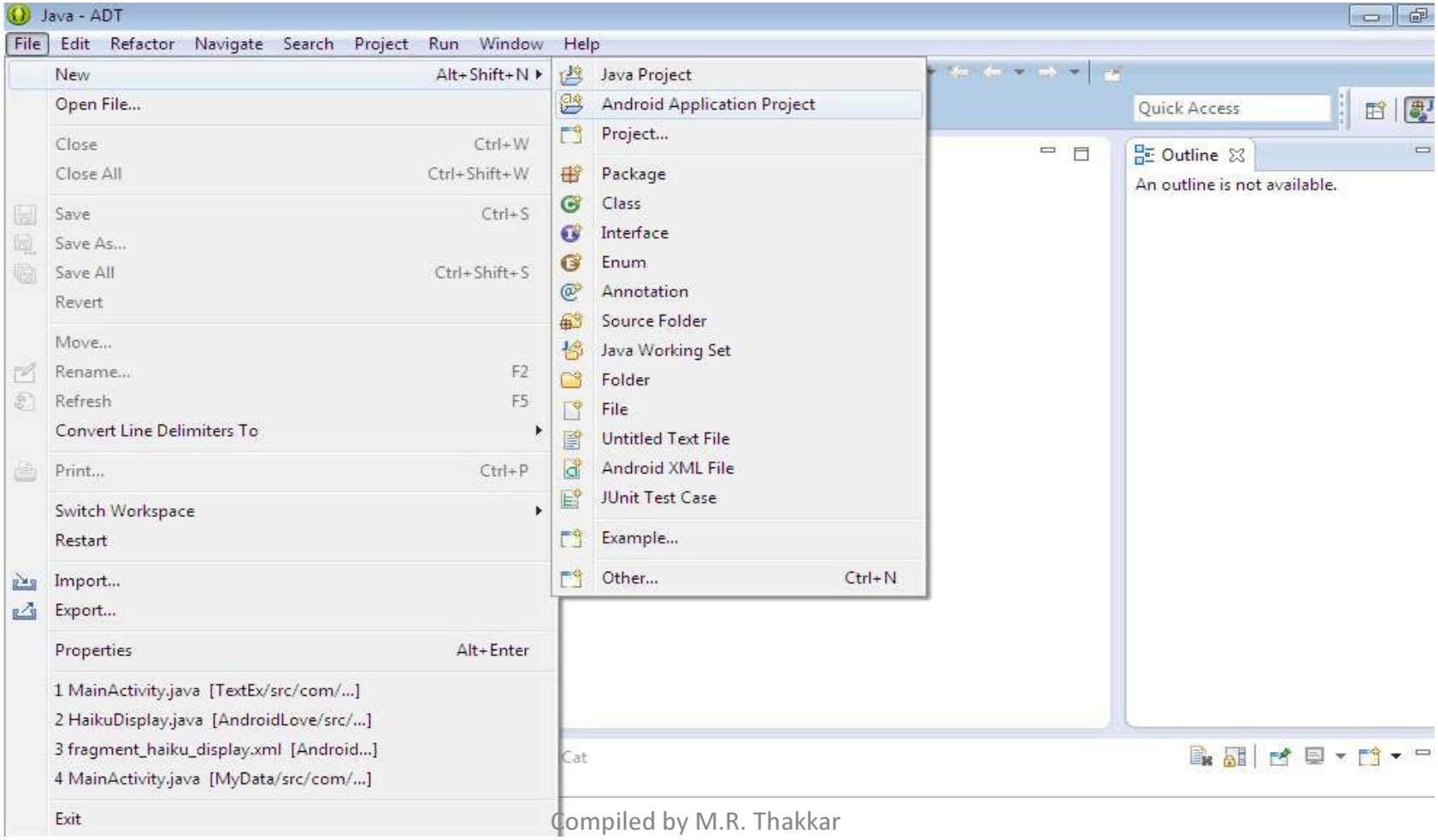
2.8 Android Emulation – Creation and set up

2.9 Android Project Framework

- Before we start our first Android Hello World application, its time to understand the Android project framework and examine all the parts that make everything work.
- First, note the various files and folders in Project Explorer that make up an Android project.
  - **src**
  - **gen**
  - **Android 4.0 library**
  - **assets**
  - **bin**
  - **AndroidManifest.xml**

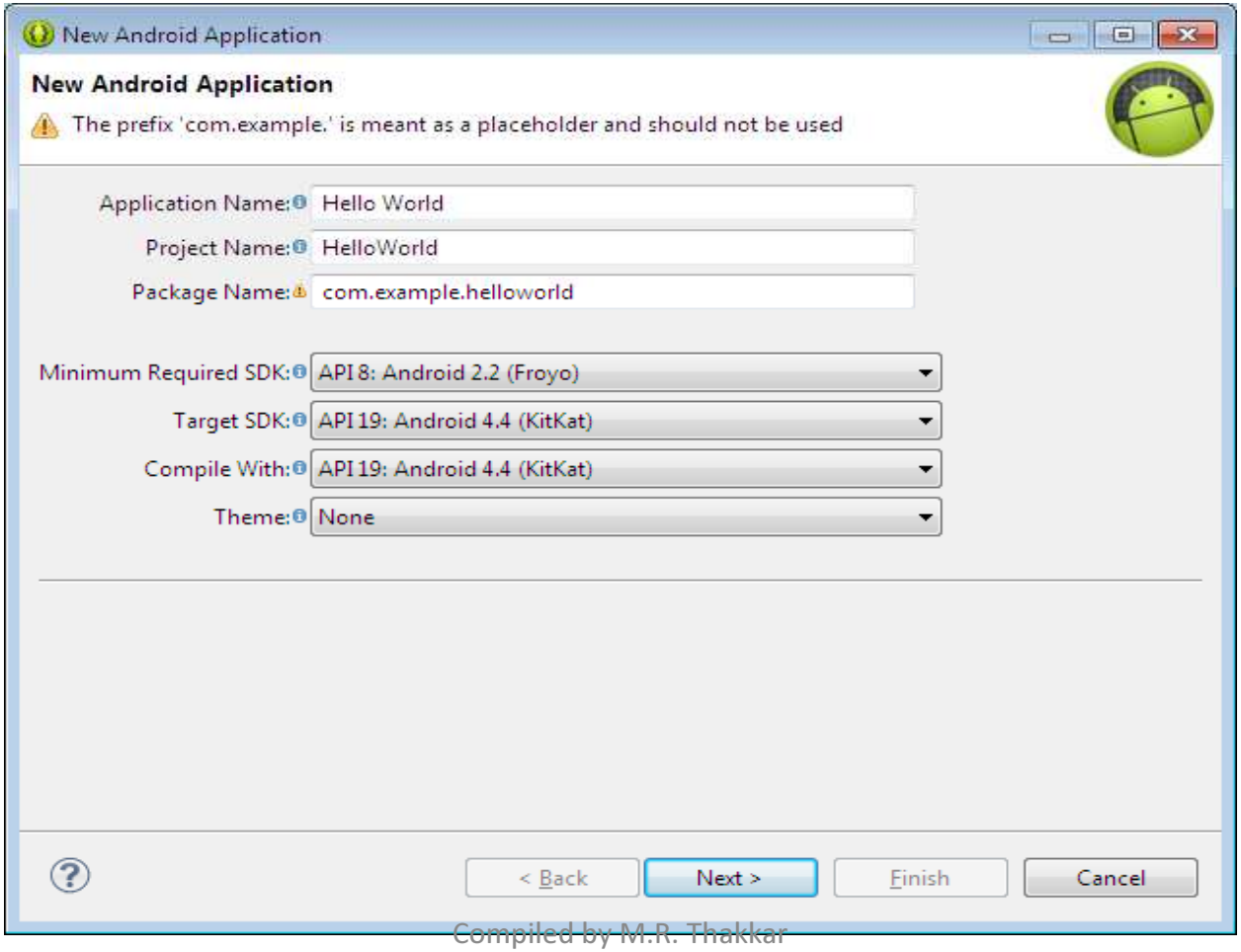
2.10 First Android Application

- **STEP 1:** Using Eclipse, create a new project by selecting **File ➪ New ➪ Android Application Project** . . .



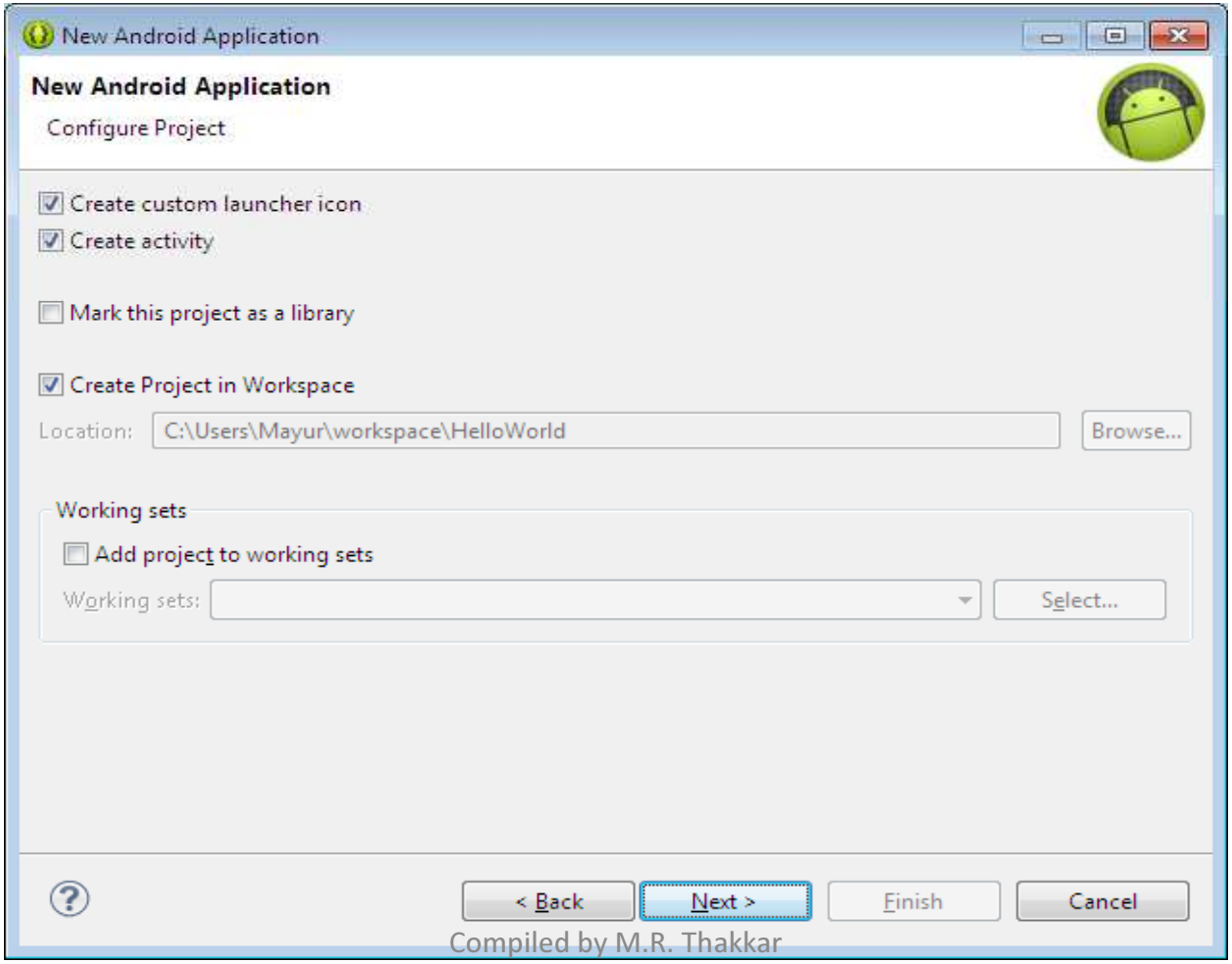
2.10 First Android Application

- **STEP 2:** In New Android Application dialog, enter **Application Name** as Hello World and click on **Next** Button…



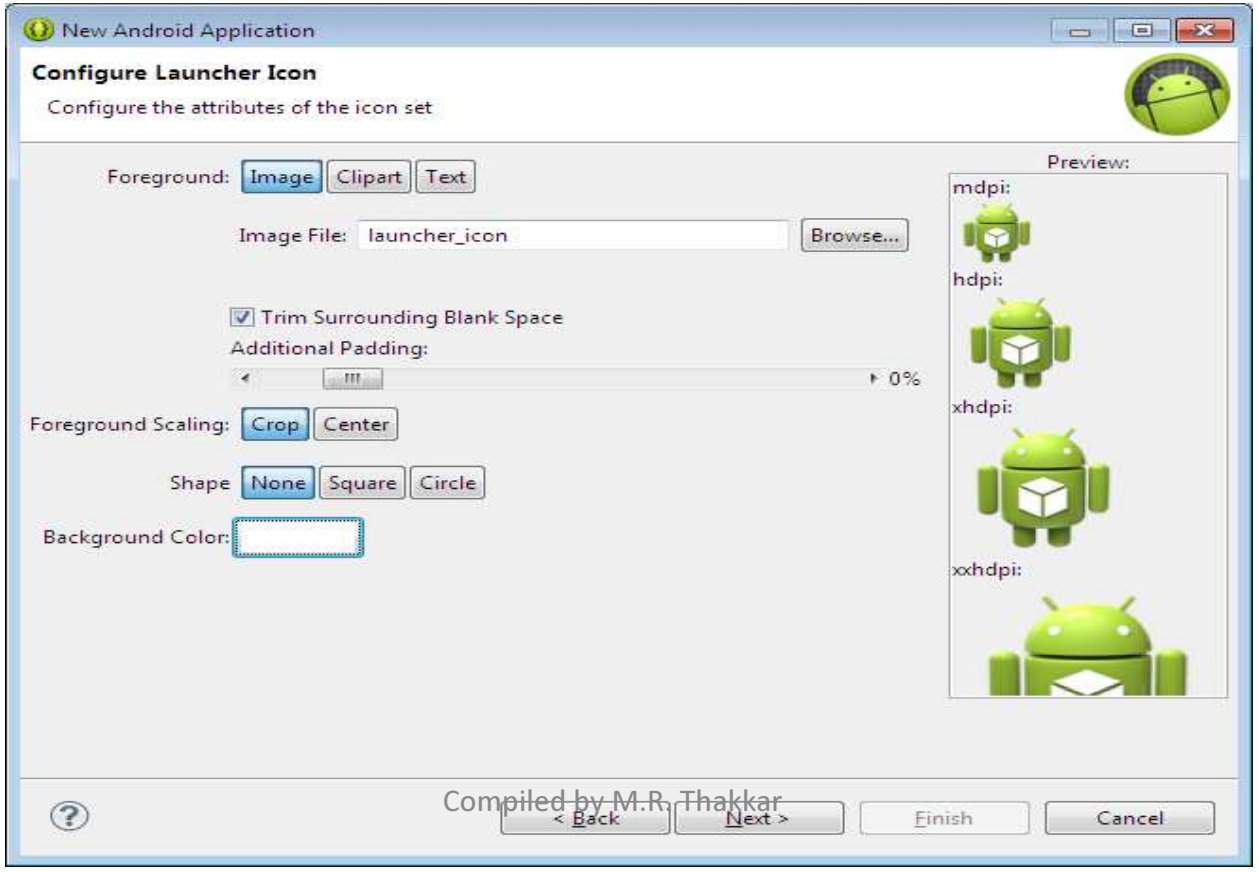
2.10 First Android Application

- **STEP 3:** In New Android Application (configure project) dialog, click on **Next** Button….



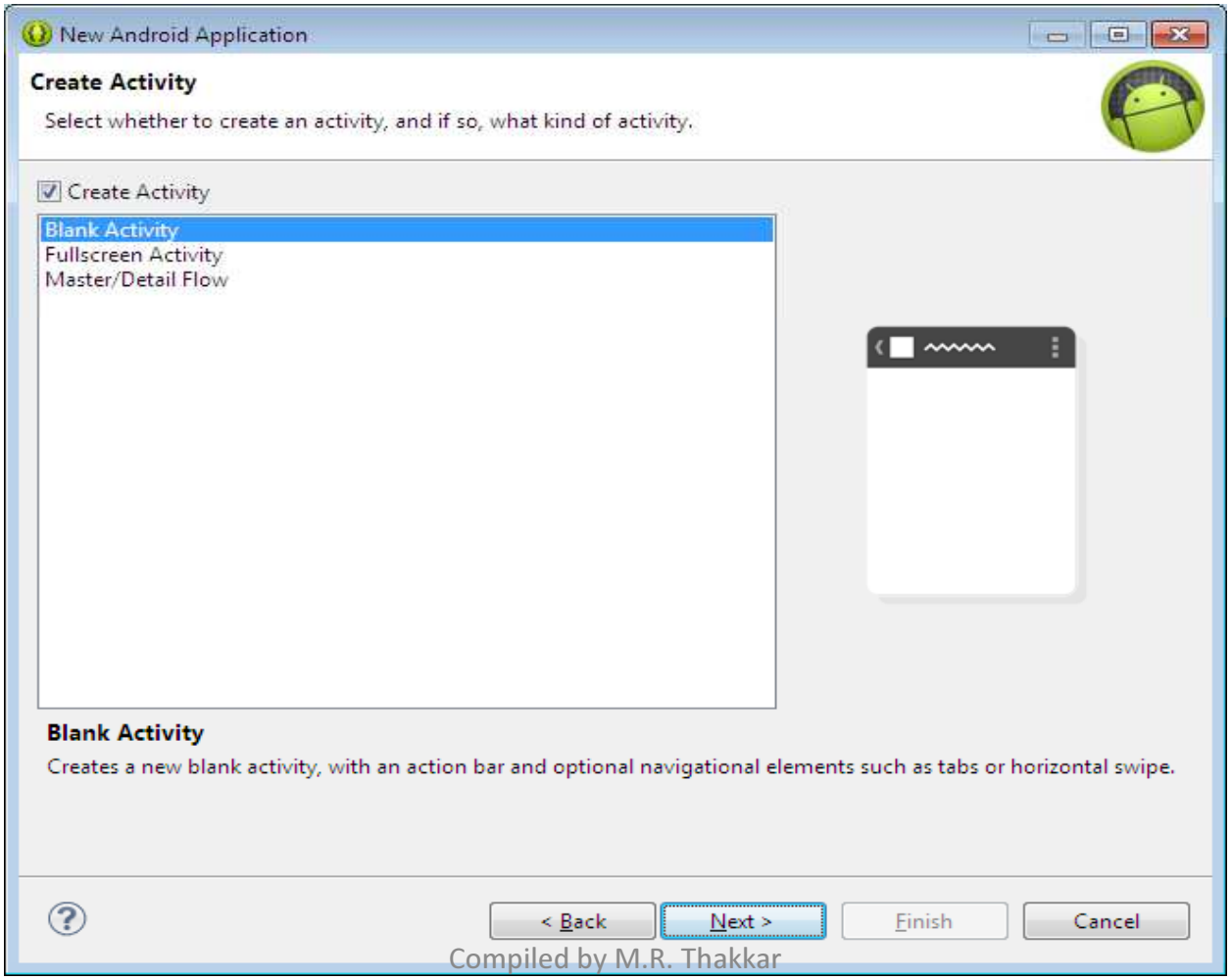
2.10 First Android Application

- **STEP 4:** In New Android Application (configure Launcher Icon) dialog, you can specify your own launching icon by providing path of image in Image File after making click on **Browse** Button, along with that you can also specify **Foreground scaling**, **Shape** and **Background color** of the application icon. After selecting appropriate setting click on **Next** button..



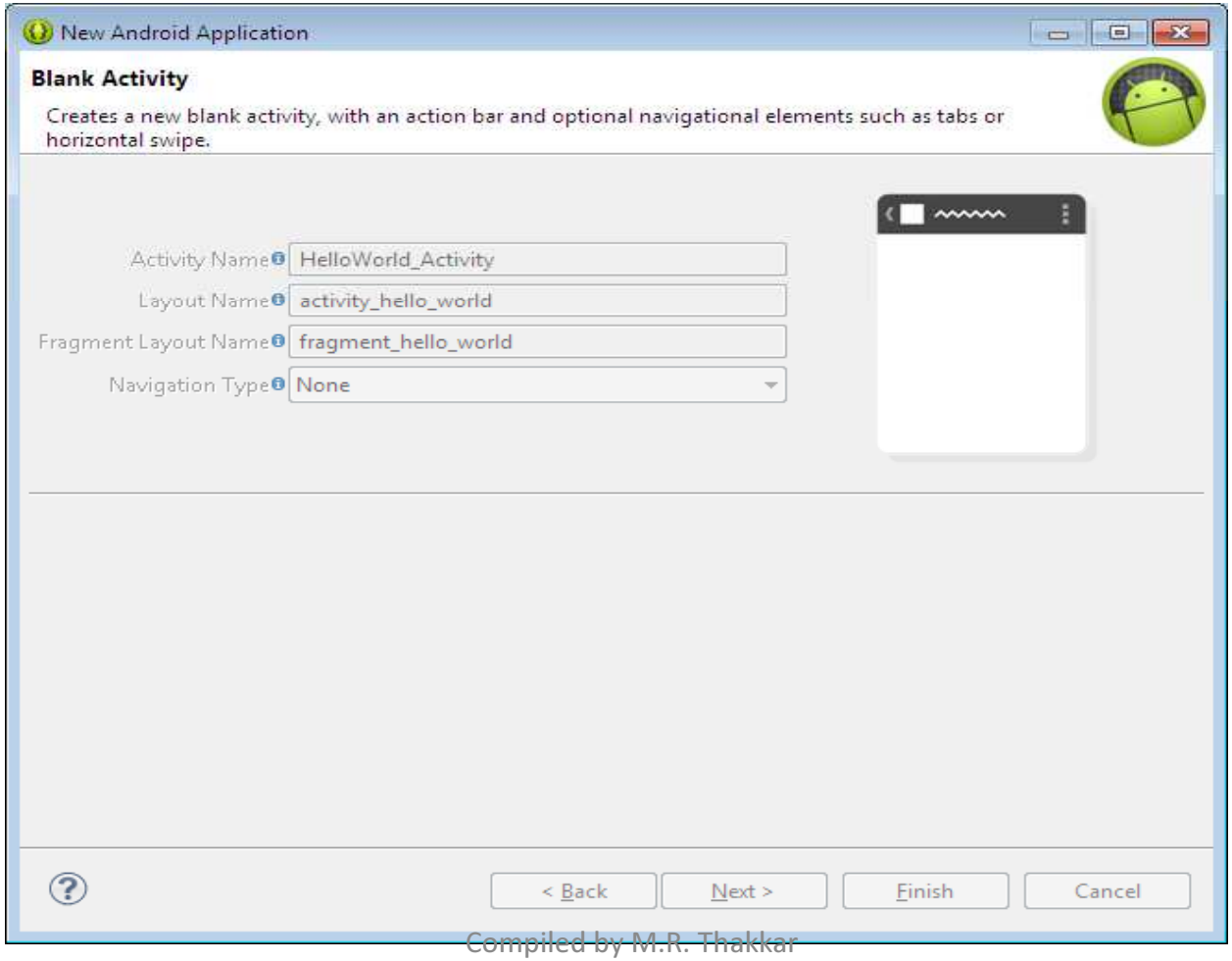
2.10 First Android Application

- **STEP 5:** In New Android Application (Create Activity) dialog, select **Blank Activity** and click on **Next** Button.



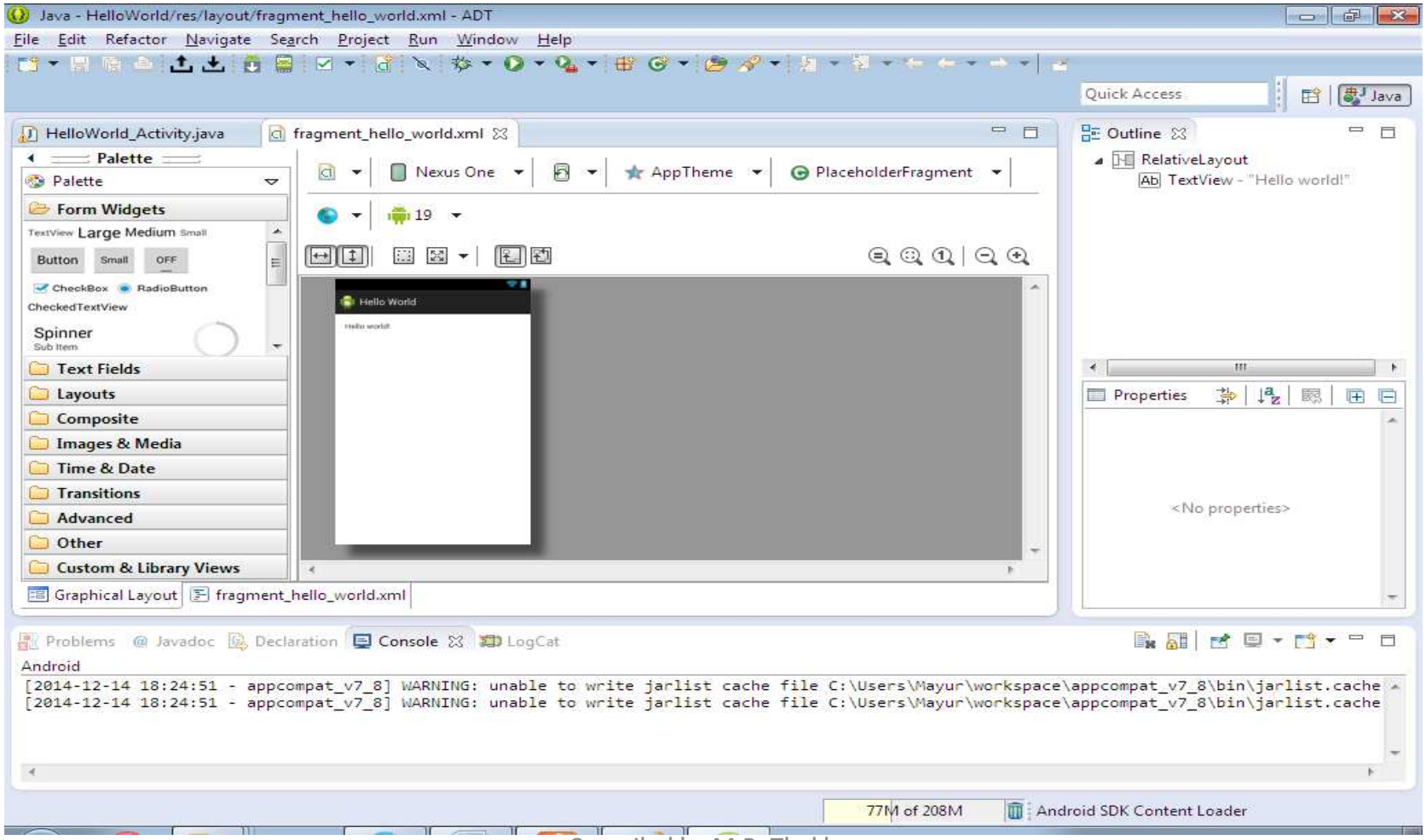
2.10 First Android Application

- **STEP 6:** In New Android Application (Blank Activity) dialog, enter **Activity Name** as HelloWorld_Activity and click on **Finish** Button.



2.10 First Android Application

- **STEP 7:** The Eclipse IDE should now look like below figure.



2.10 First Android Application

- It is recommended that you store all the string constants in your application in this strings.xml file and reference these strings using the @string identifier as android:text attribute is specified in TextView. The strings.xml fi le located in the res/values folder.

- **strings.xml**

```
<?xml version="1.0" encoding="utf-8"?>

<resources>
<string name="app_name">App1</string>
<string name="action_settings">Settings</string>
<string name="hello_world">Hello world!</string>
</resources>
```

2.10 First Android Application

- The layout file activity_main.xml defines the user interface (UI) of your application. The default view is the Graphical Layout view, which lays out the activity graphically. To modify the UI by hand, click the activity_main.xml tab. It will show you following code.

- **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:text="@string/hello_world" />

</RelativeLayout>
```

2.10 First Android Application

- You will also find another file MainActivity.java file in the src directory, inside com.example.helloworld package, where you will write the code of your application.

MainActivity.java

```
package com.example.helloworld;

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

public class MainActivity extends ActionBarActivity
 {

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