Programming in Android

Session: 2

Getting Started with Android

Objectives

- Explain the process of creating an Android application
- Explain the fundamentals of Android application
- Explain the composition of Android applications framework
- Explain communication components
- Explain pre-existing components

Introduction

- A developer needs to understand the basics of Android Applications
- A supported IDE needs to be used –
  - Android Studio
  - Eclipse IDE
  - IntelliJ
- Applications can be deployed on AVDs or real devices

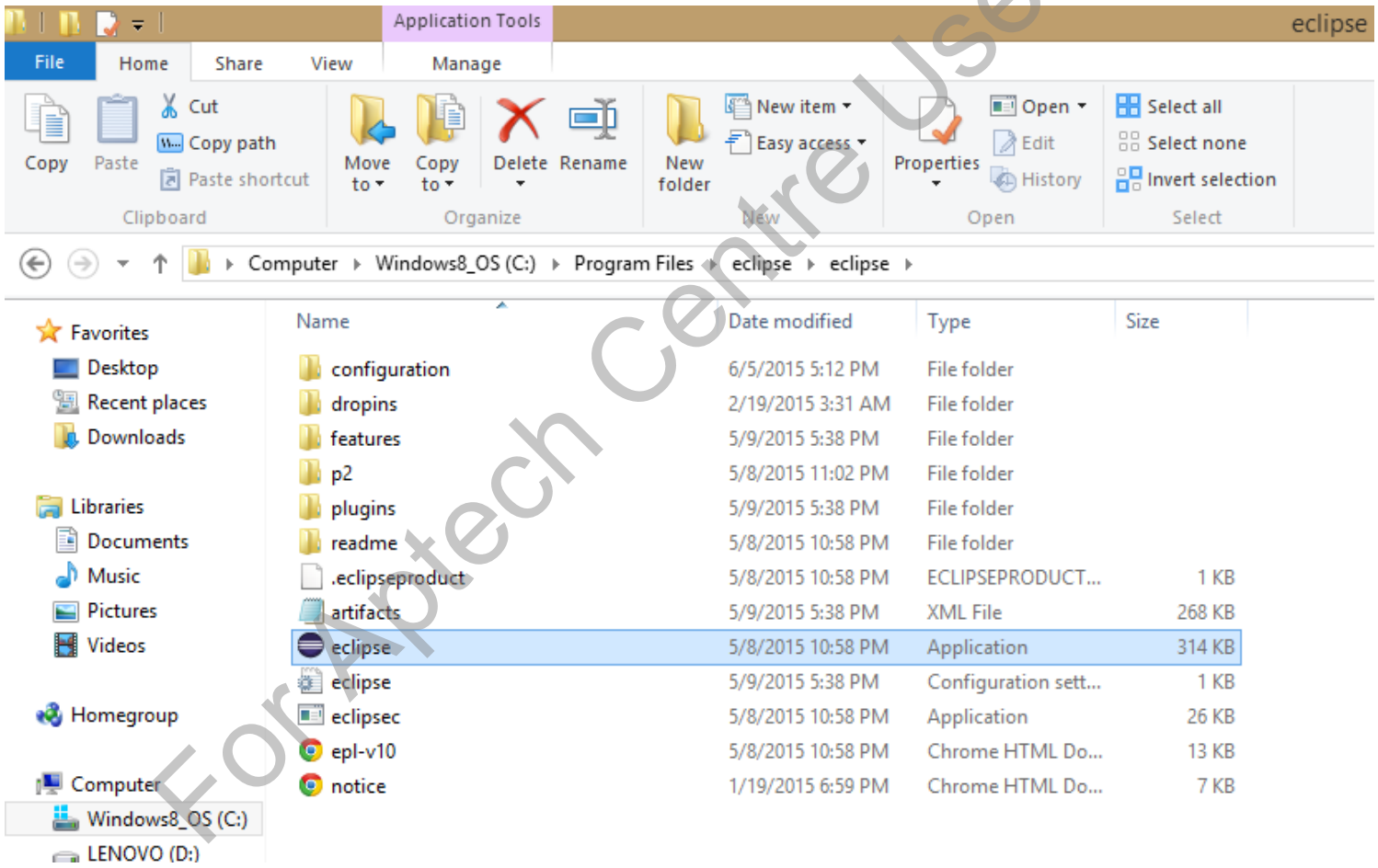
Creating an Android Application

- What is an Android Project
  - An Android project consists of the entire source code for an Android application
  - It gets compiled and packaged into an apk

- An Android project can be created in two ways:
  - Eclipse IDE
  - Android Studio

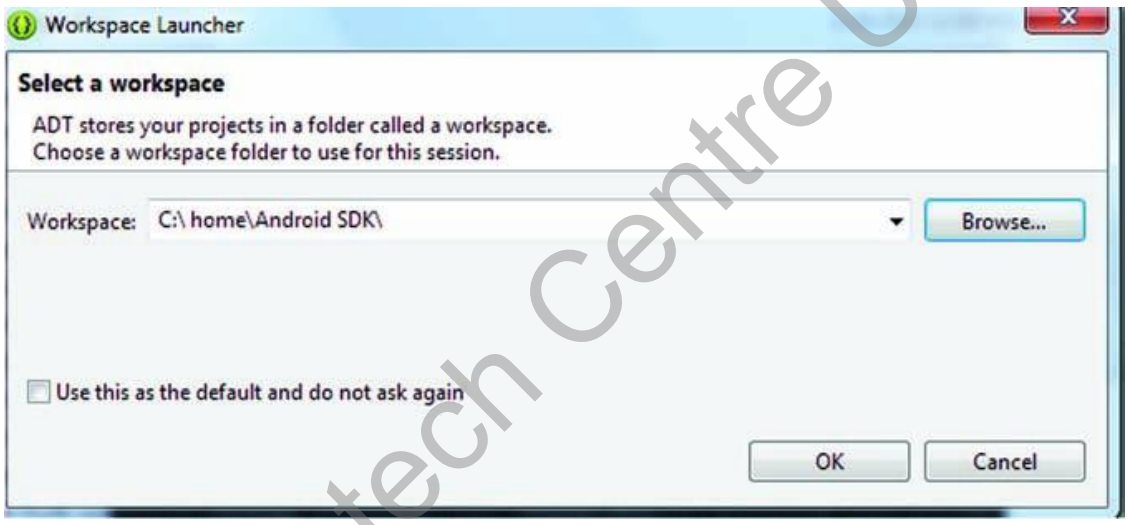
HelloWorld Android Project Using Eclipse 1-9

- Access the folder where the Eclipse has been installed/extracted
- Double-click the executable file as shown in the following figure:



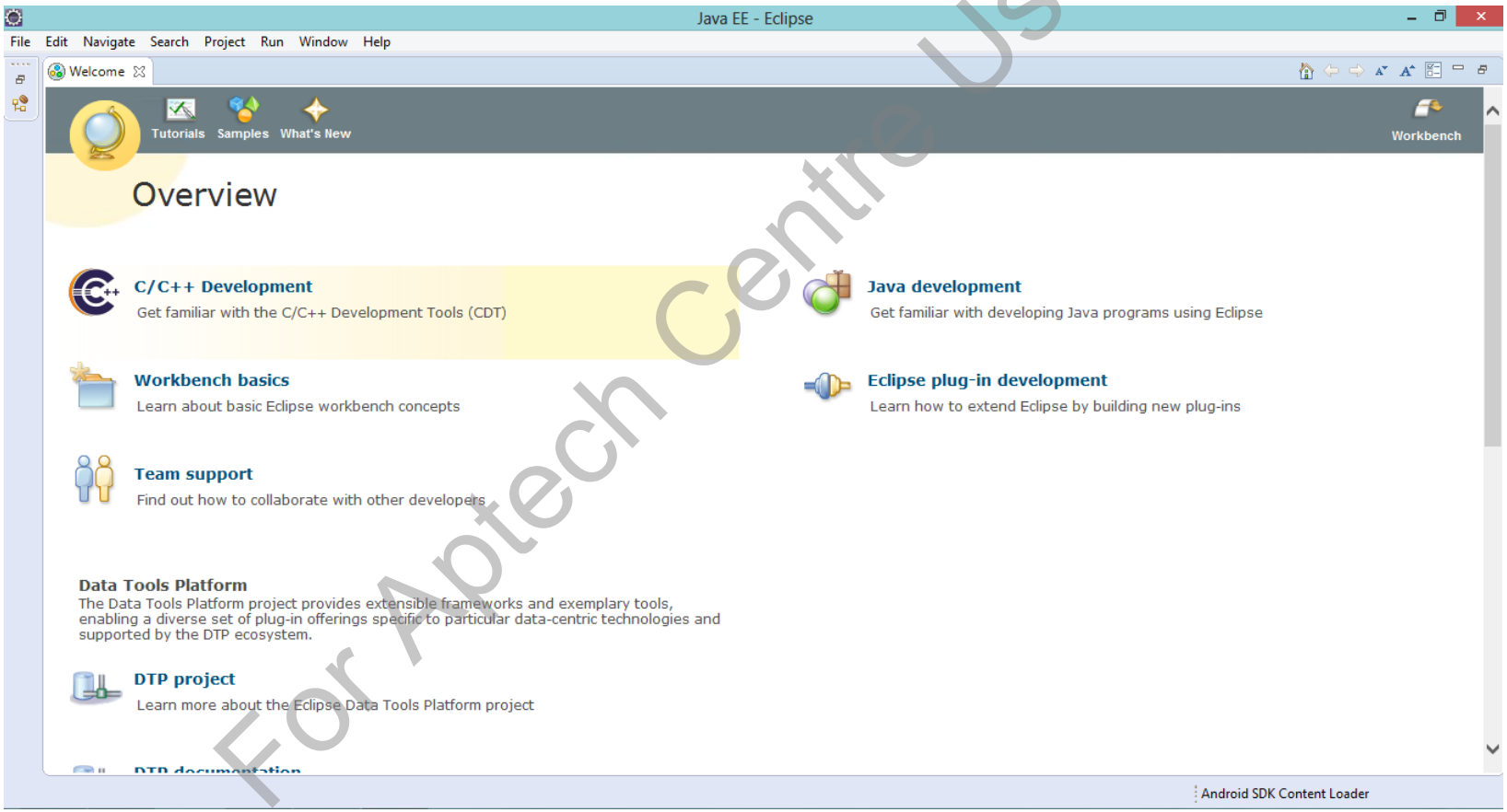
HelloWorld Android Project Using Eclipse 2-9

- In the Workspace Launcher dialog box, click Browse and select a folder where to the project needs to be stored as shown in the following figure:



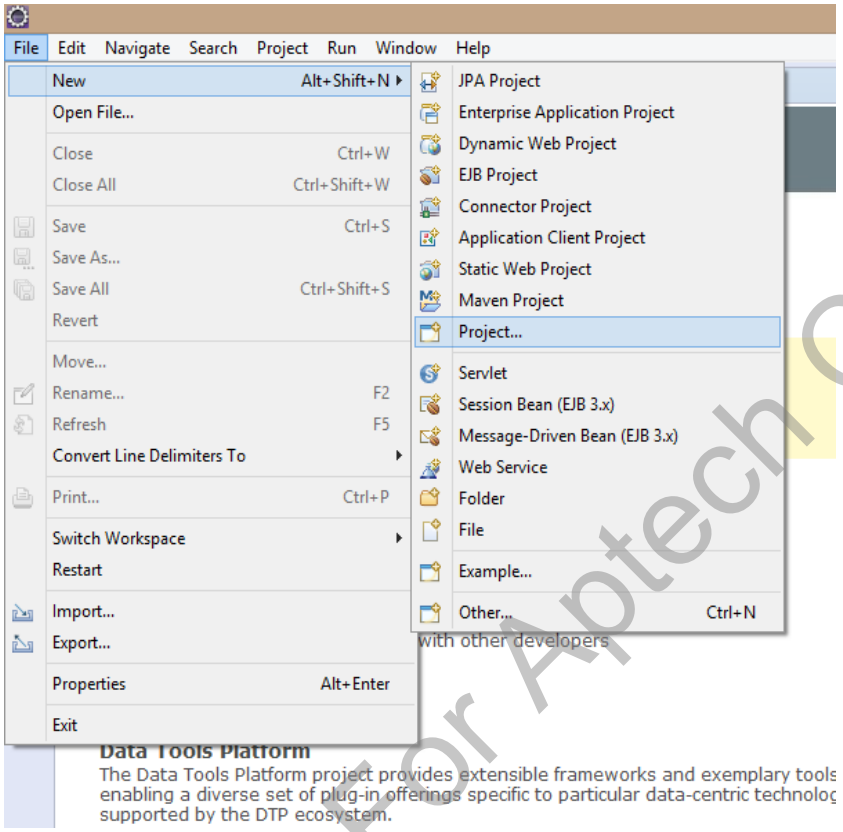
HelloWorld Android Project Using Eclipse 3-9

- Click OK. Within Eclipse IDE, the Home Screen is as shown in the following figure:

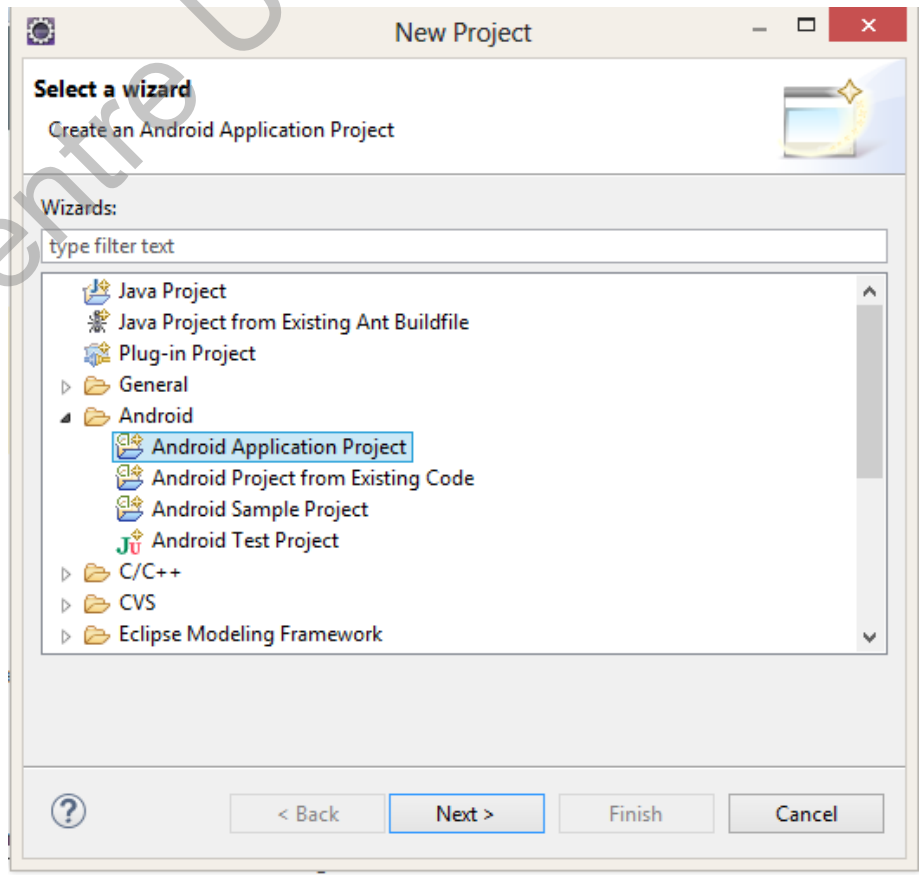


HelloWorld Android Project Using Eclipse 4-9

- Click File → New → Project as shown in the following figure:

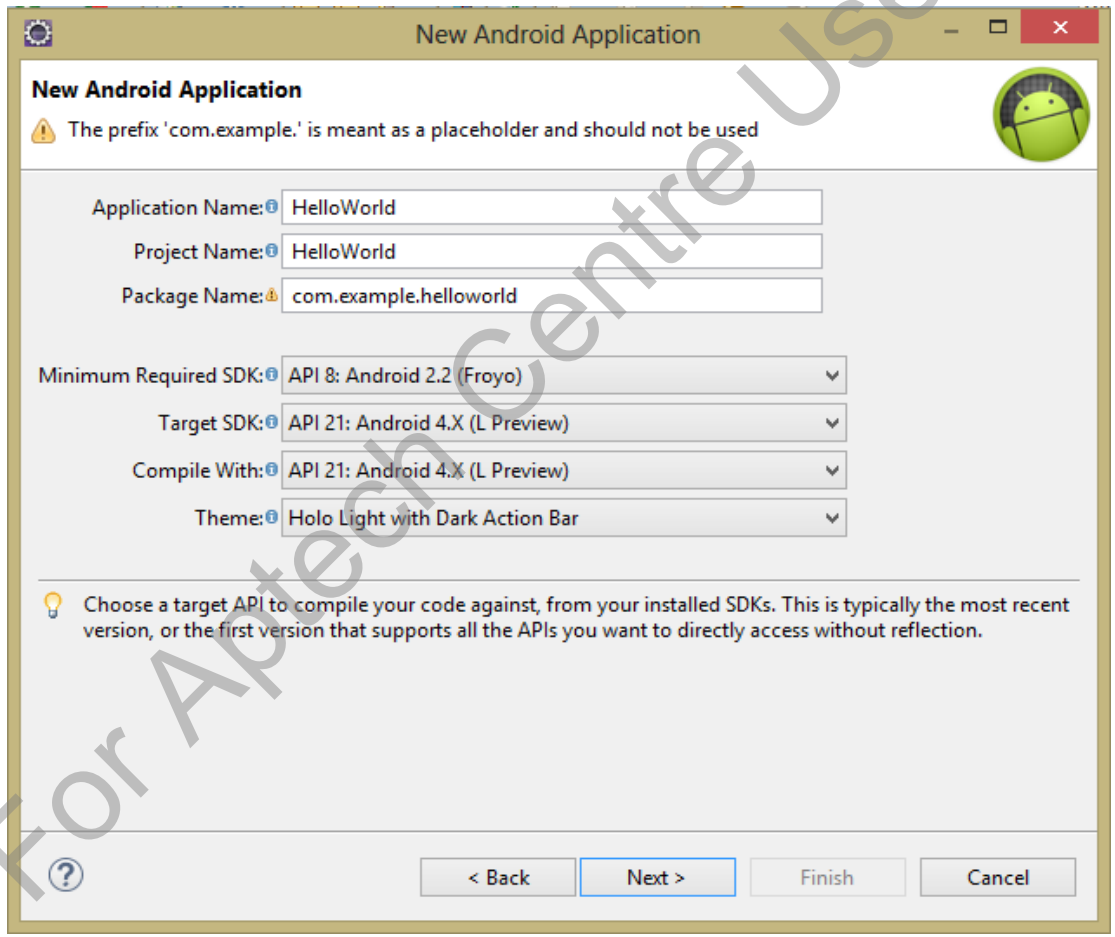


- From the new Wizard that appears, select Android → Android Application and click Next as shown in the following figure:



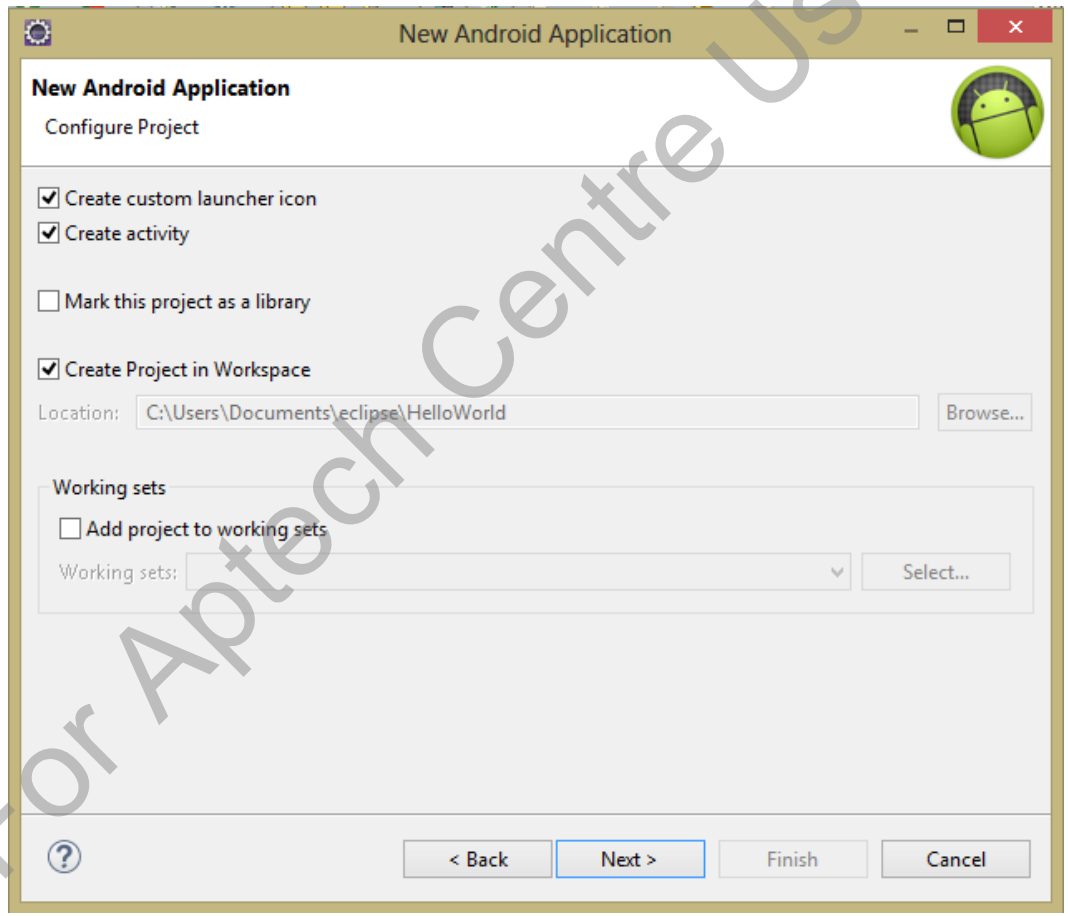
HelloWorld Android Project Using Eclipse 5-9

- In the Application Name box, type HelloWorld as shown in the following figure:



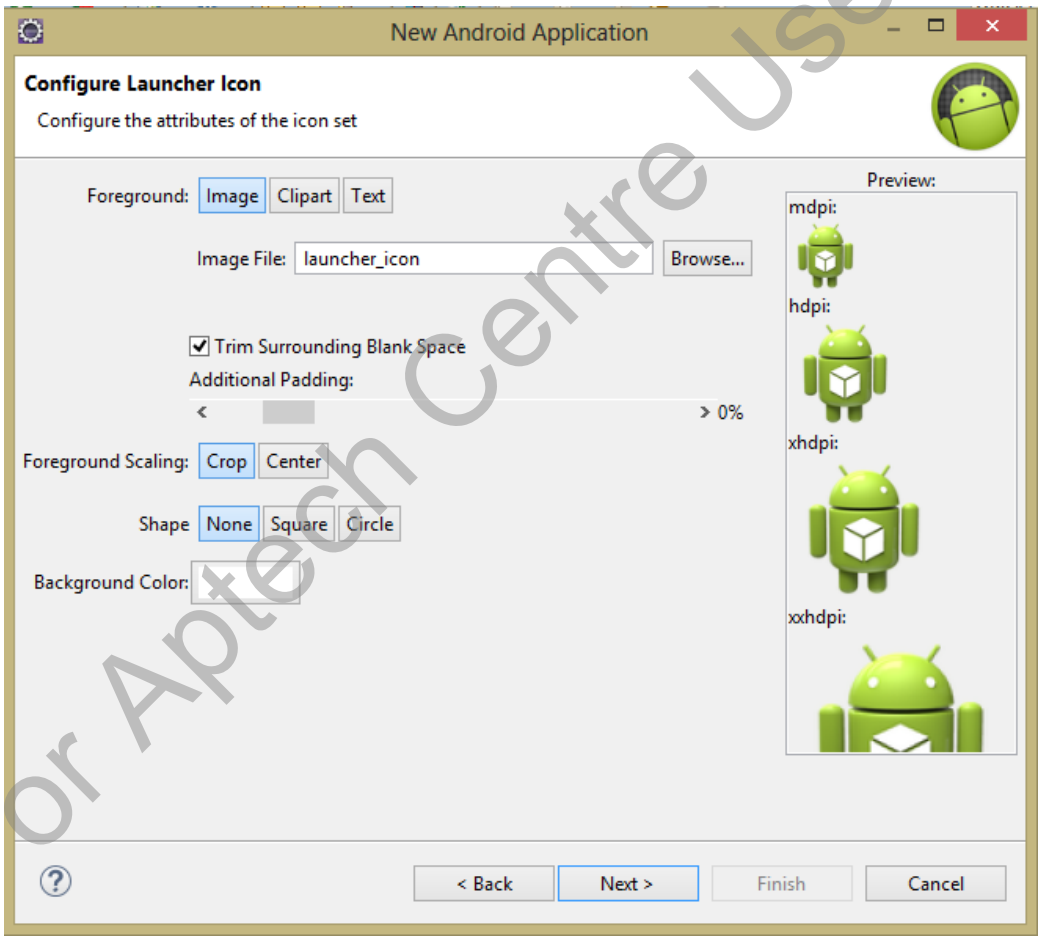
HelloWorld Android Project Using Eclipse 6-9

- Click Next twice in the Configure Project pane, as shown in the following figure:



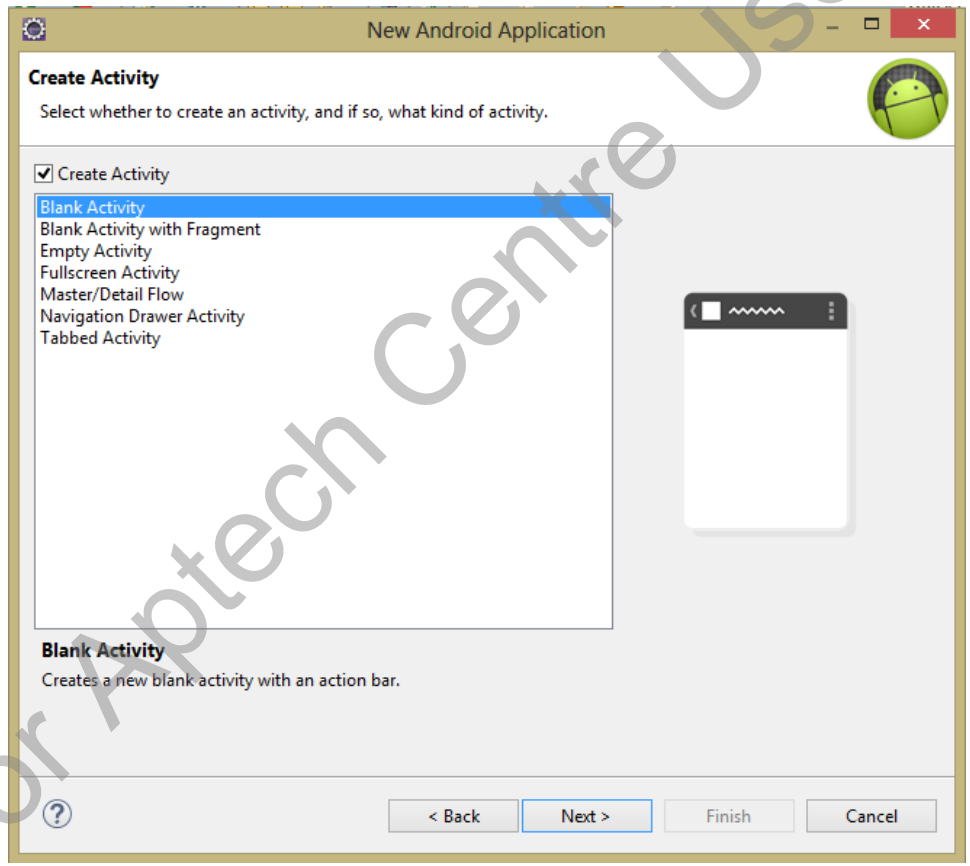
HelloWorld Android Project Using Eclipse 7-9

- In the Configure Launcher Icon screen that is displayed, customize the appearance of the launcher icon as shown in the following figure:



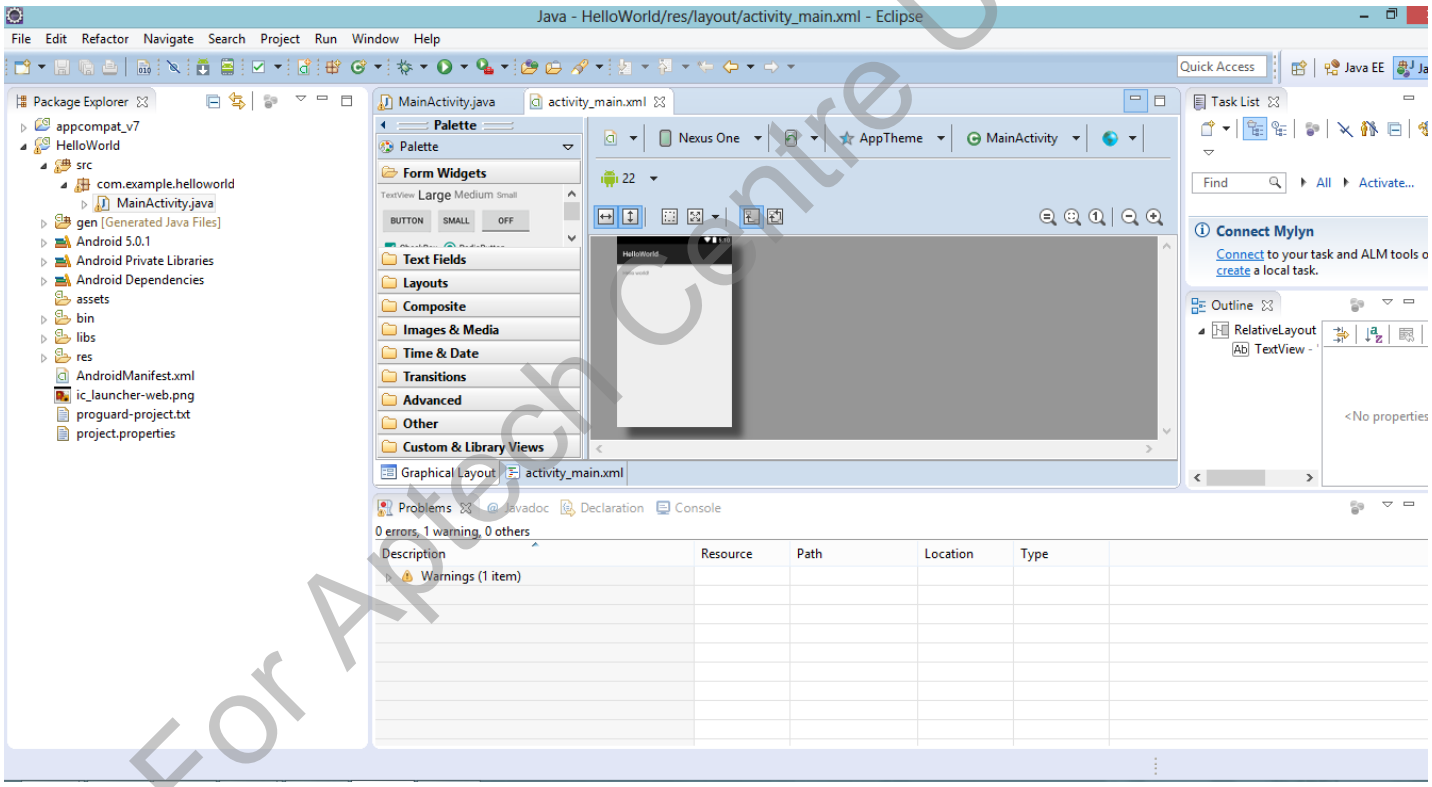
HelloWorld Android Project Using Eclipse 8-9

- Click Next. The Create Activity screen is displayed as shown in the following figure:



HelloWorld Android Project Using Eclipse 9-9

- Select Blank Activity and click Next.
- Click Finish. The project is created as shown in the following figure:

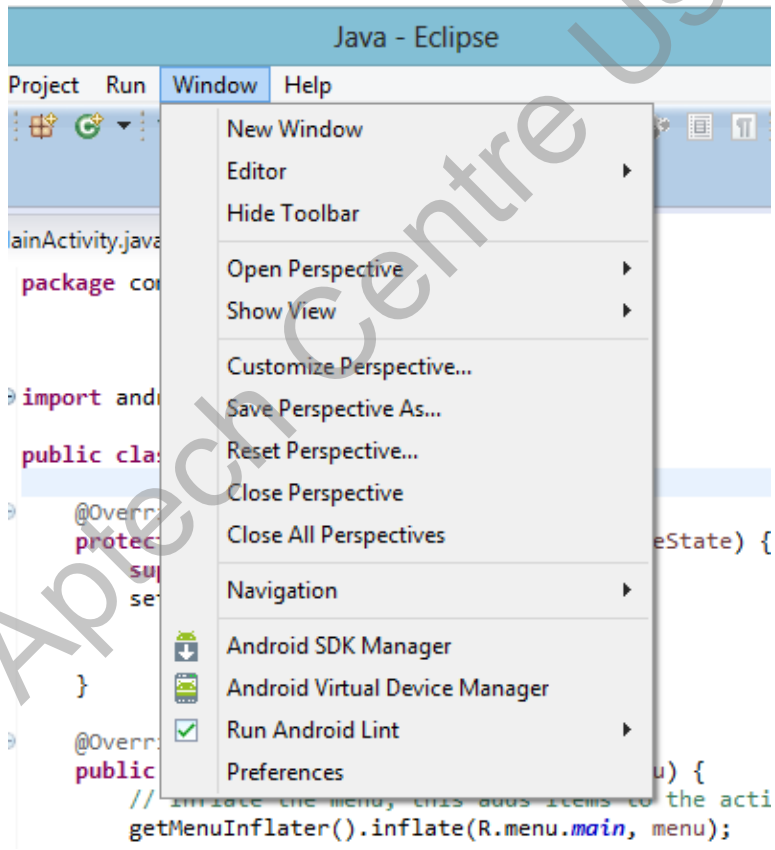


Android Virtual Device

- An Android Virtual Device is an Android Emulator
- The emulator functions exactly similar to a mobile device
- AVD consists of:
  - Hardware features
  - Software features
  - The look of the screen

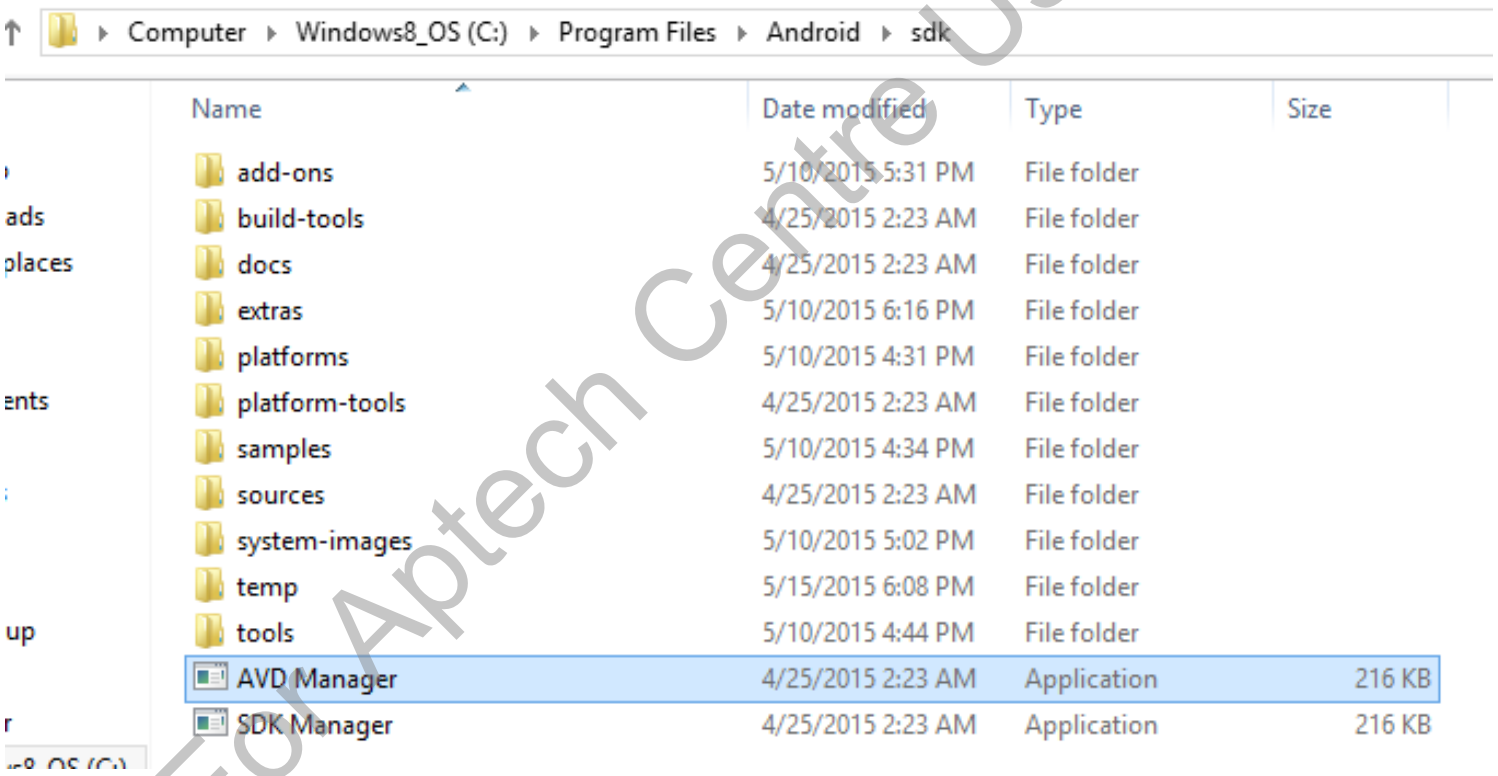
Creating an AVD 1-6

- Click Window → Android Virtual Device Manager to start the AVD Manager as shown in the following figure:



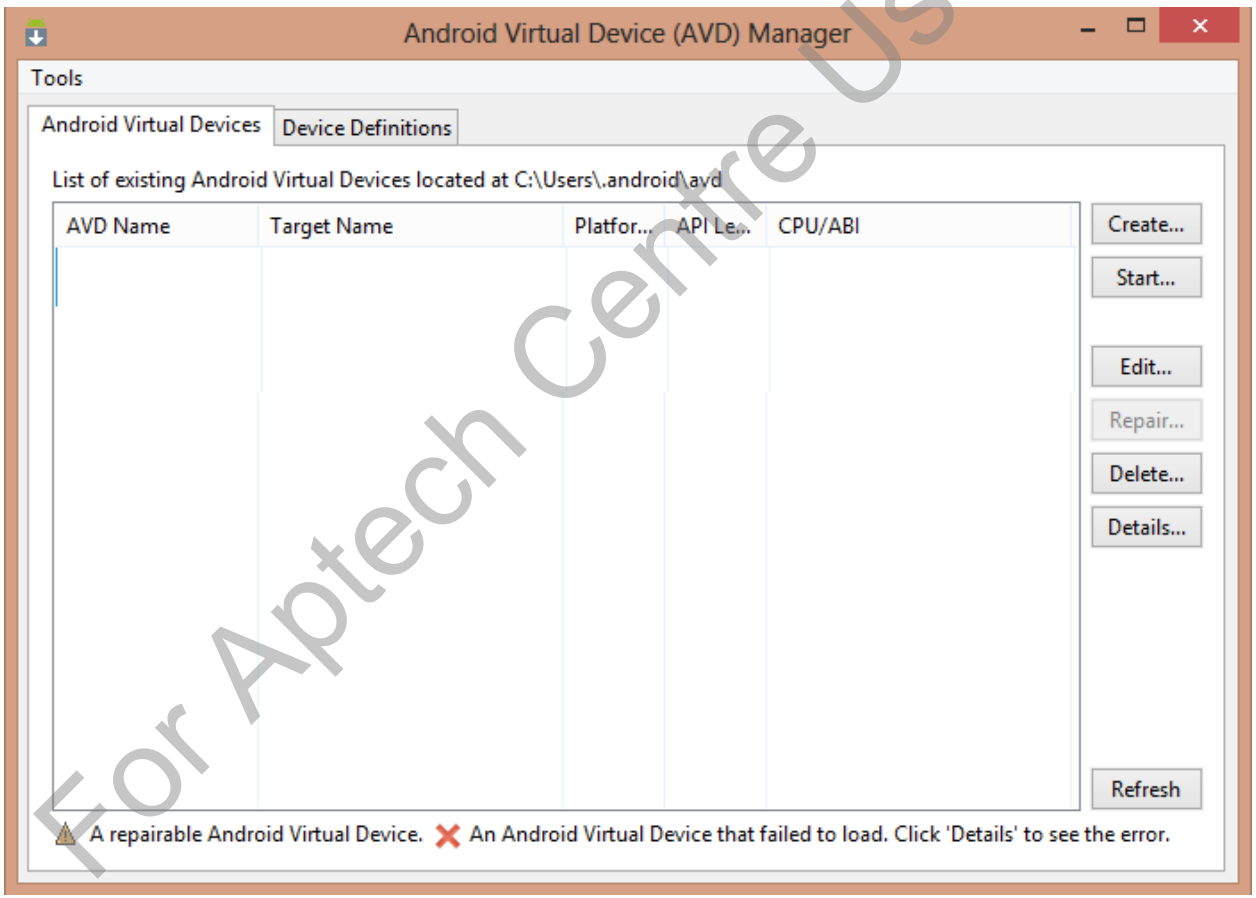
Creating an AVD 2-6

- Alternately, navigate to the Android SDK installation directory and start the AVD Manager as shown in the following figure:



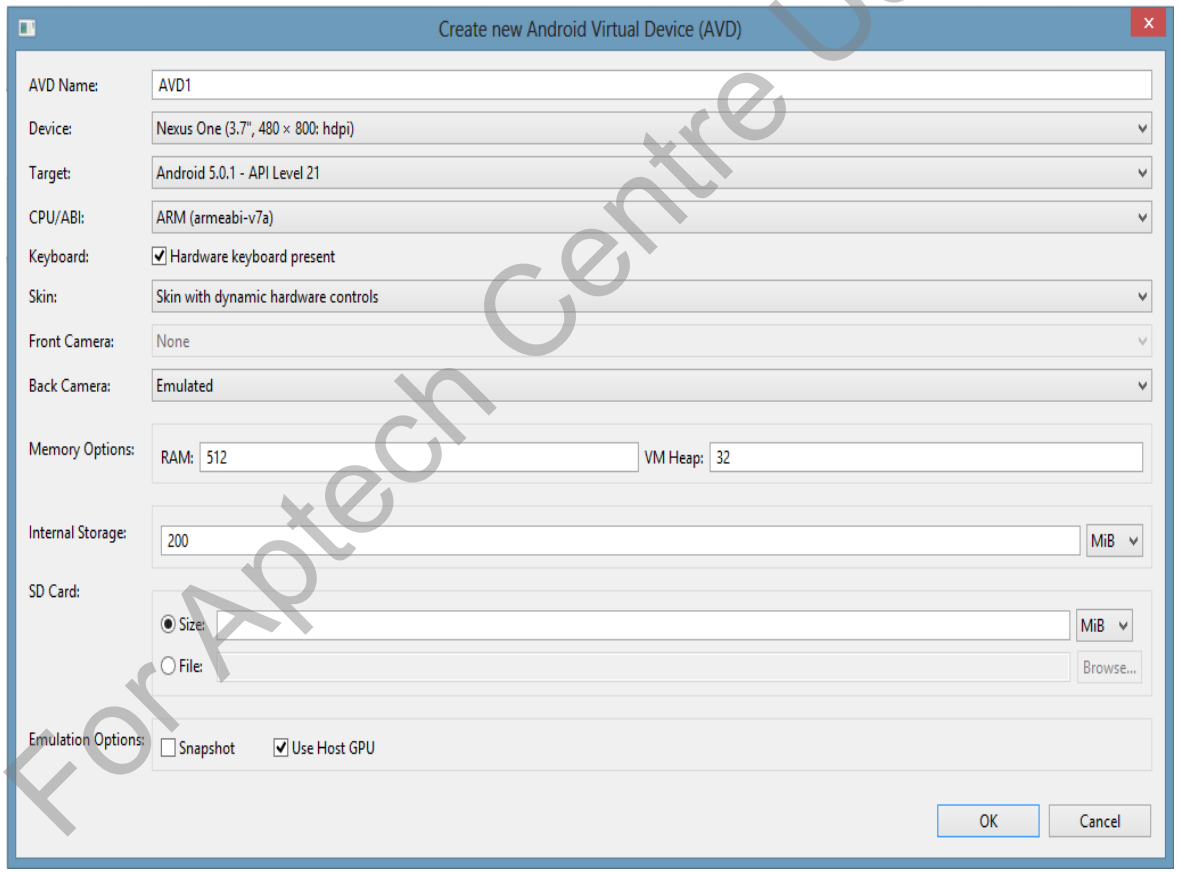
Creating an AVD 3-6

- The Android Virtual Device (AVD) Manager dialog box is displayed as shown in the following figure:



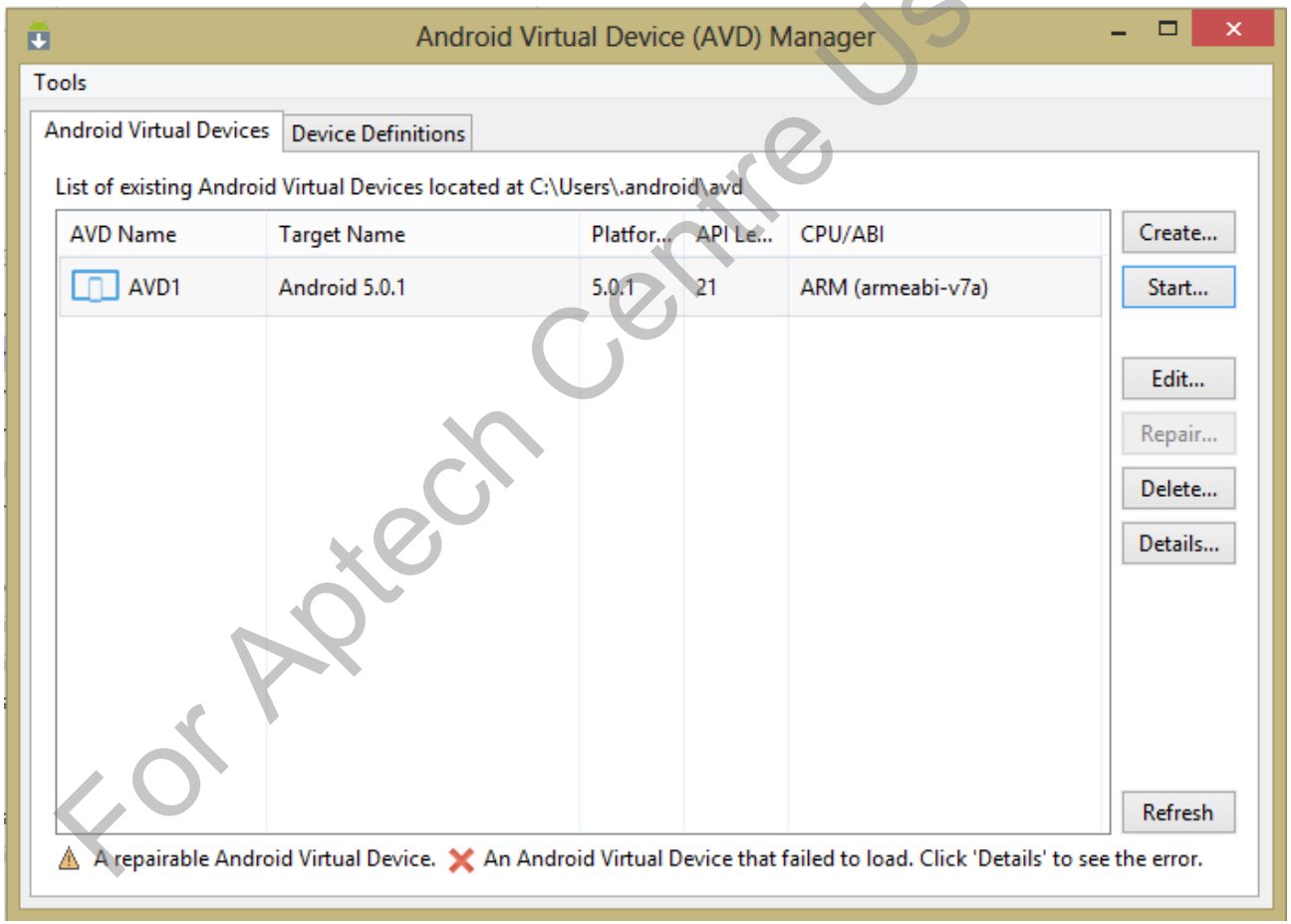
Creating an AVD 4-6

- Click Create to create a new AVD. The Create new Android Virtual Device (AVD) dialog box is displayed.
- Enter the AVD details as shown in the following figure:



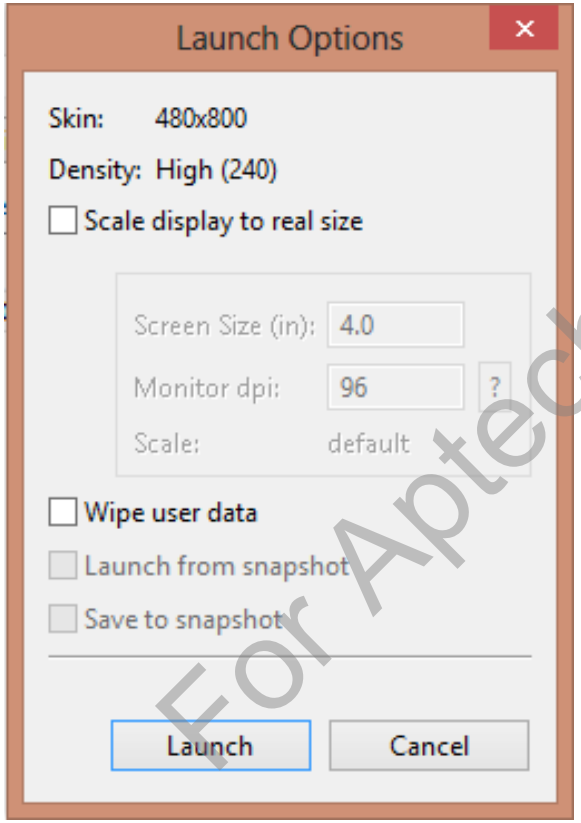
Creating an AVD 5-6

- Click OK after all the requirements are complete.
- The Android Virtual Device Manager dialog box appears displaying the newly created AVD, as shown in the following figure:



Creating an AVD 6-6

- Select AVD1 and click Start to start the emulator.
- Click Launch in the Launch Options dialog box as shown in the following figure:



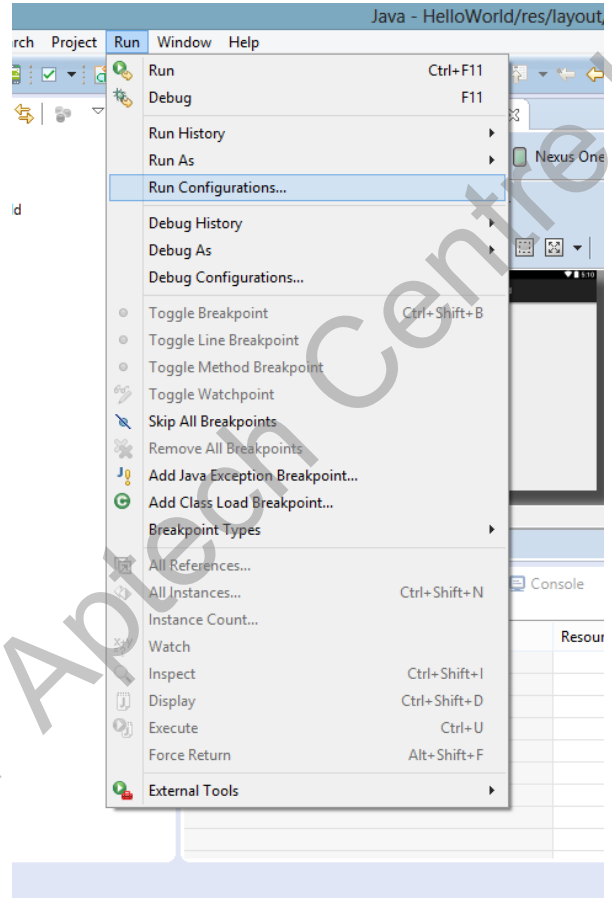
- The emulator is launched as shown in the following figure:

Creating Launch Configurations

- A launch configuration is required to run the app on the AVD
- There are two type of launch configurations –
  - Run Configurations
  - Debug Configurations

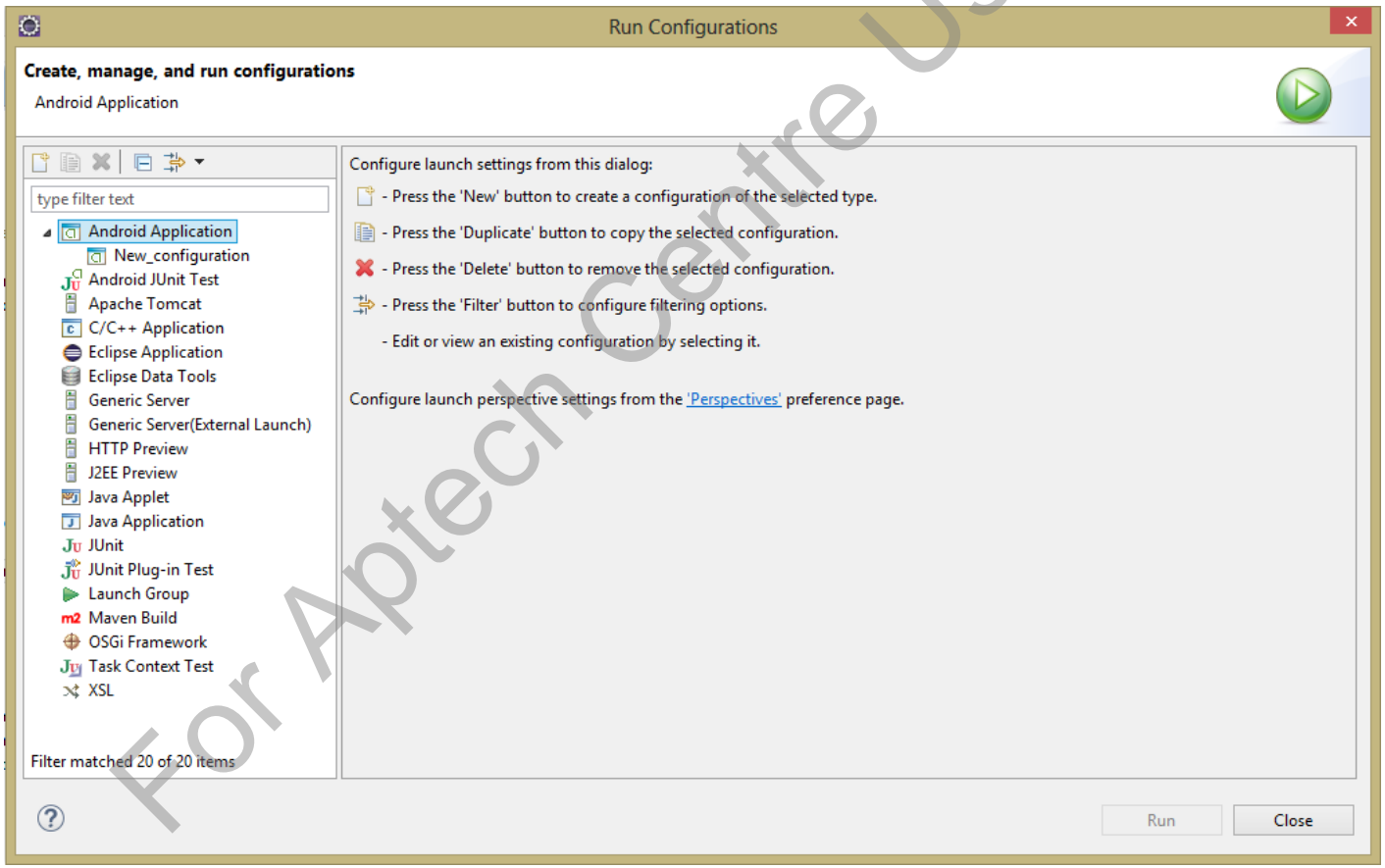
Creating Run Configurations 1-4

- In Eclipse, click Run → Run Configurations as shown in the following figure:



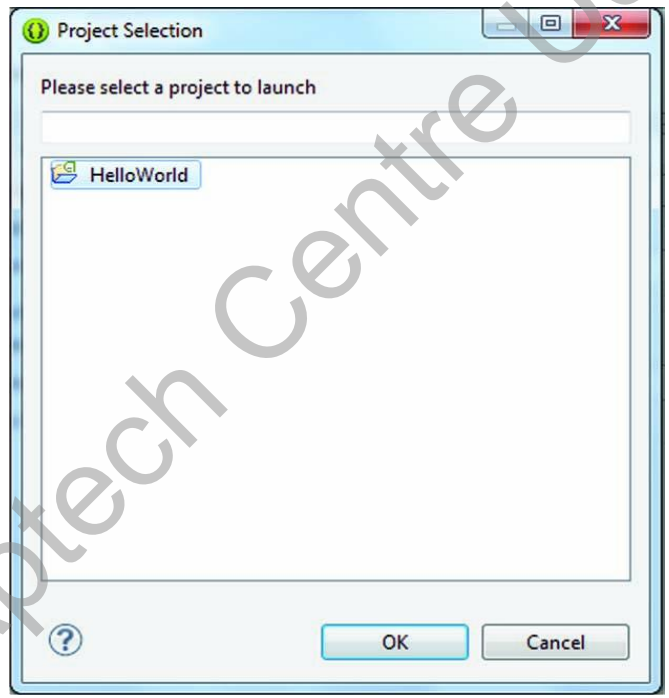
Creating Run Configurations 2-4

- In the Create, manage, and run configurations dialog box, select Android Application and click New launch configuration icon as shown in the following figure:



Creating Run Configurations 3-4

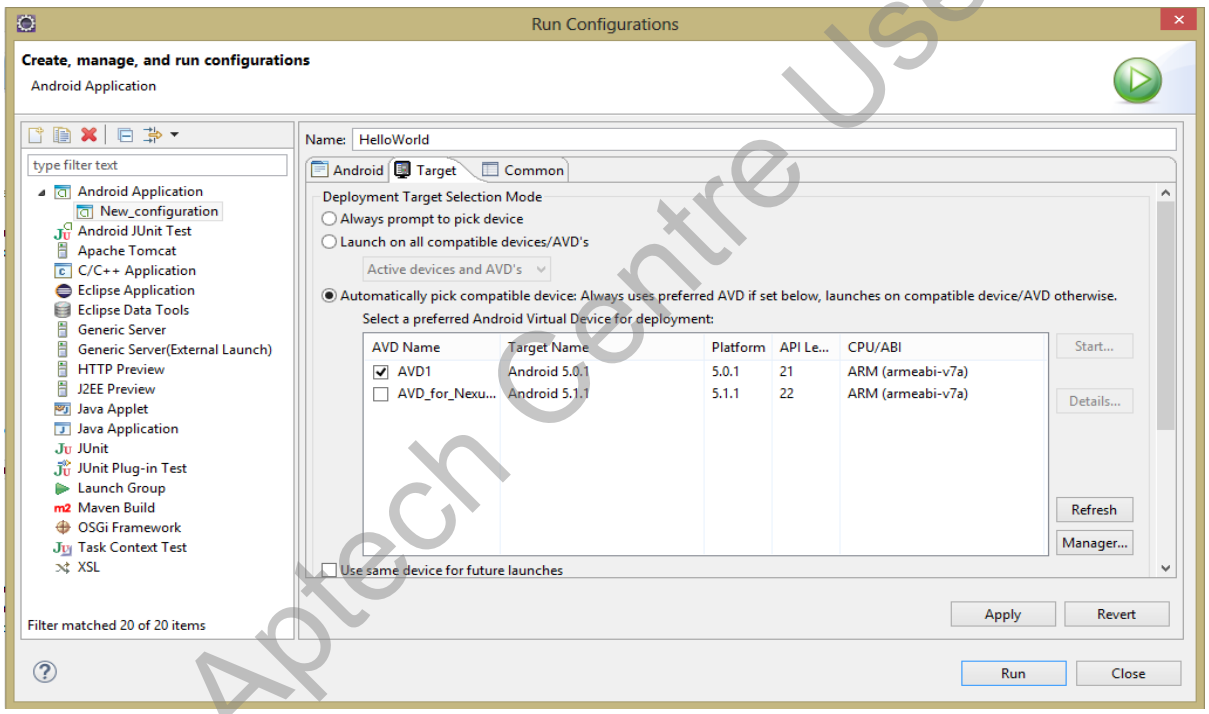
- In the Name box type the name of the new configuration as Hello World
- Click Browse to display the Project Selection dialog box as shown in the following figure:



- Select HelloWorld and click OK

Creating Run Configurations 4-4

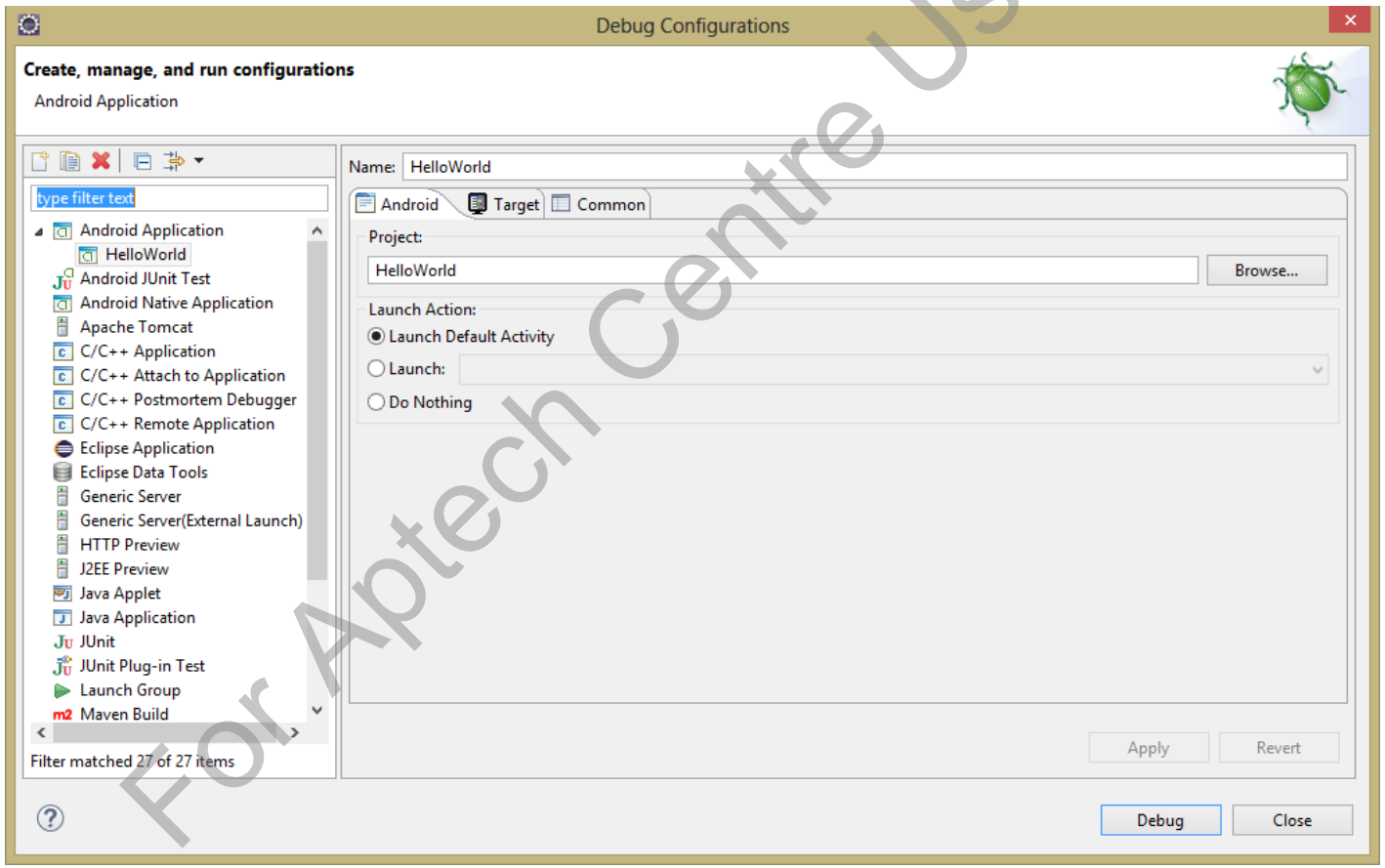
- Click the Target tab and select the newly created AVD, as shown in the following figure:



- Click Apply
- Click Close to exit the Create, manage, and run configurations dialog box

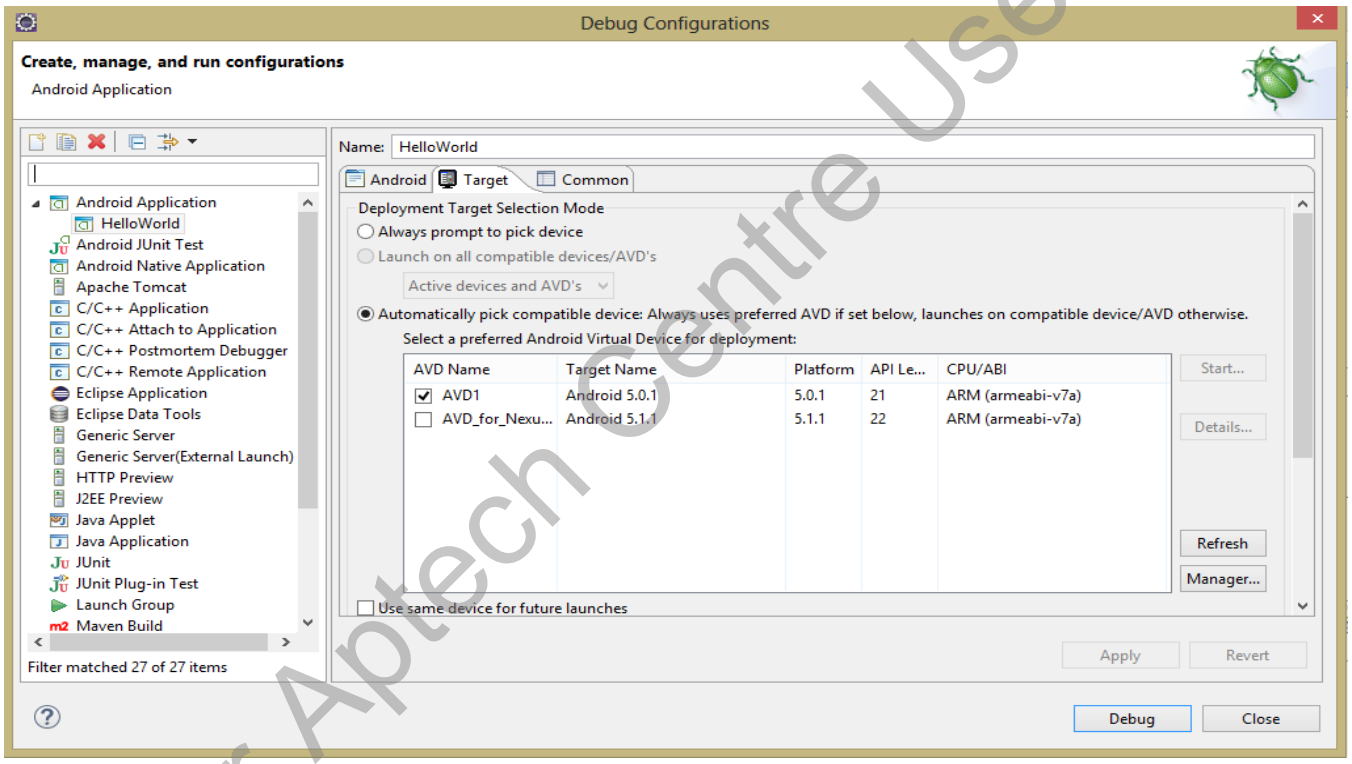
Creating Debug Configurations 1-2

- In Eclipse, click Run Debug → Configurations
- In the Debug Configurations dialog box, in the Android Application list, select HelloWorld as shown in the following figure:



Creating Debug Configurations 2-2

- Click the Target tab. The AVD for the configuration is selected by default as shown in the following figure:



- Click Debug
- Click Run → Run

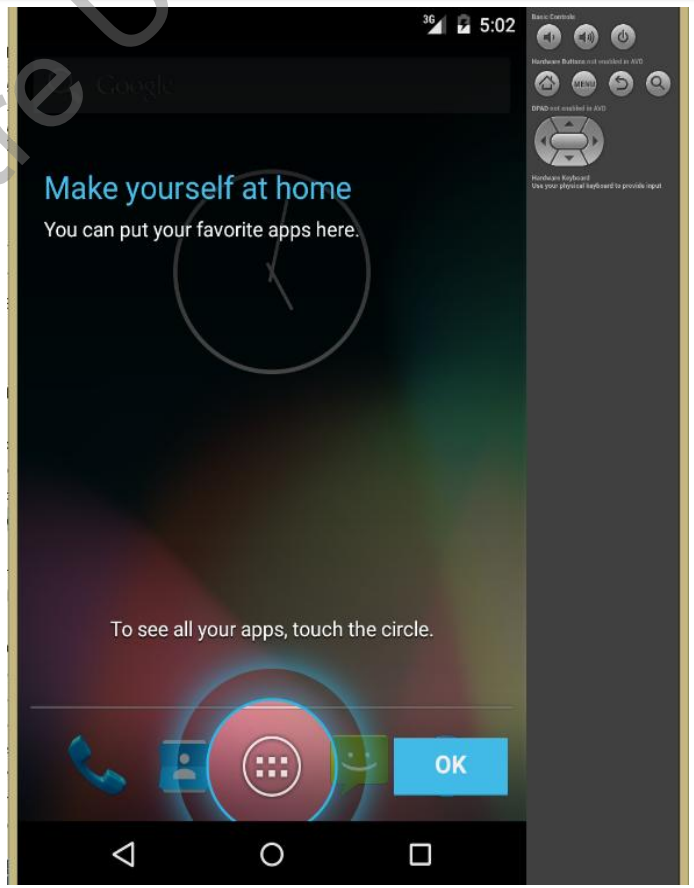
Running and Debugging HelloWorld Project

- The apk is installed on the target
- Running and Debugging can be done on –
  - Real Android Device
  - Emulator
- Its deployed automatically using adb, by the IDE

Running the Project 1-2

- In Eclipse, click Run → Run
- The emulator is displayed with the lock screen as shown in the following figure:

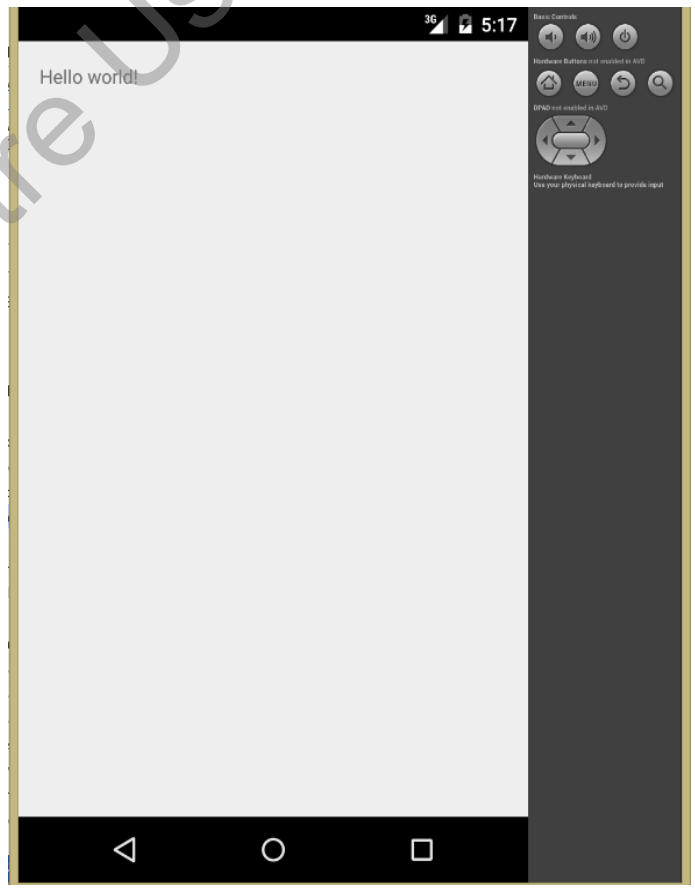
- Click and drag the lock icon upwards to unlock the home screen
- On the home screen, click the circle and click OK as shown in the following figure:



Running the Project 2-2

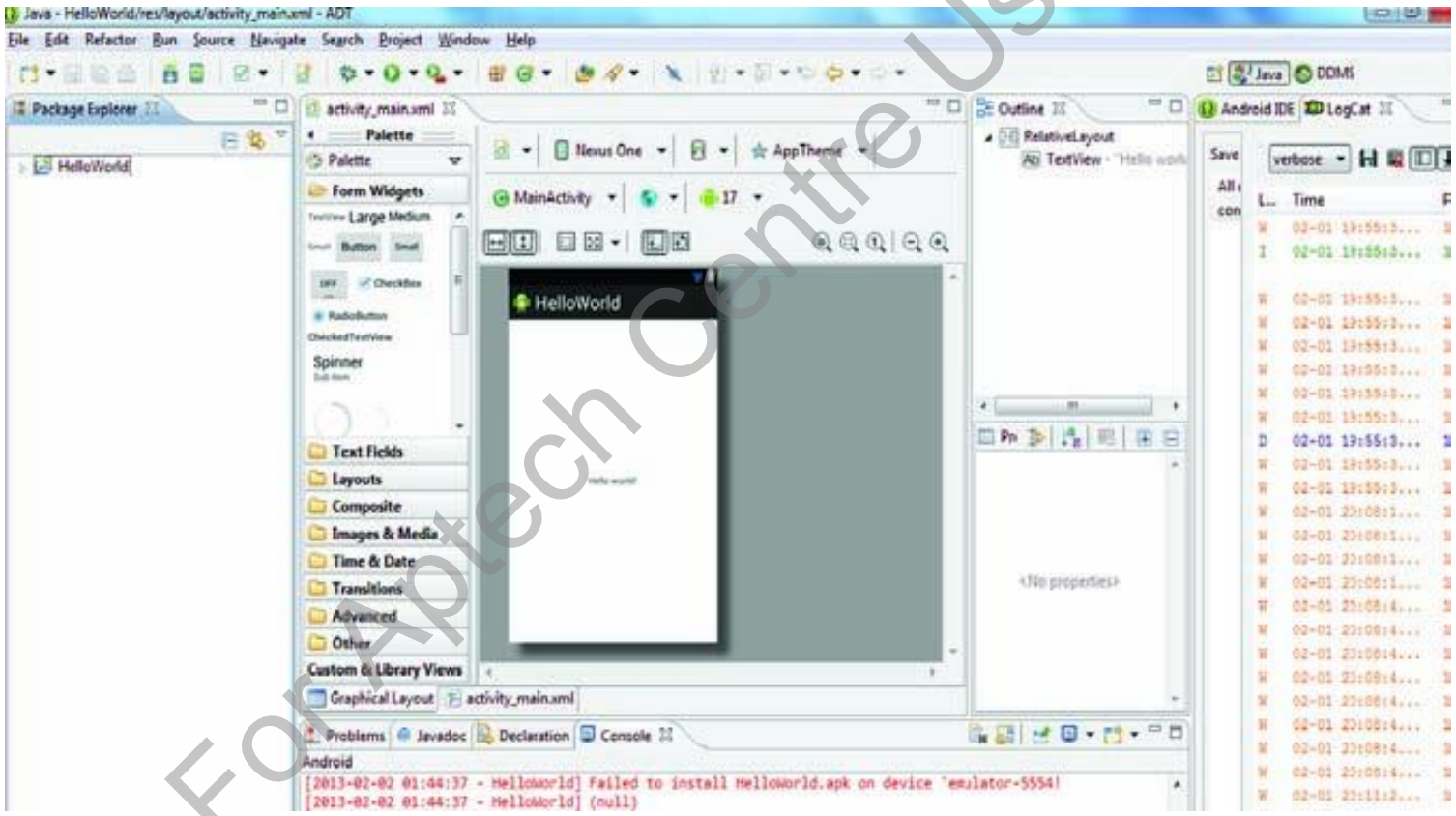
- When the apps load, click the HelloWorld app icon as shown in the following figure:

- The text Hello world! is displayed on the screen as shown in the following figure:



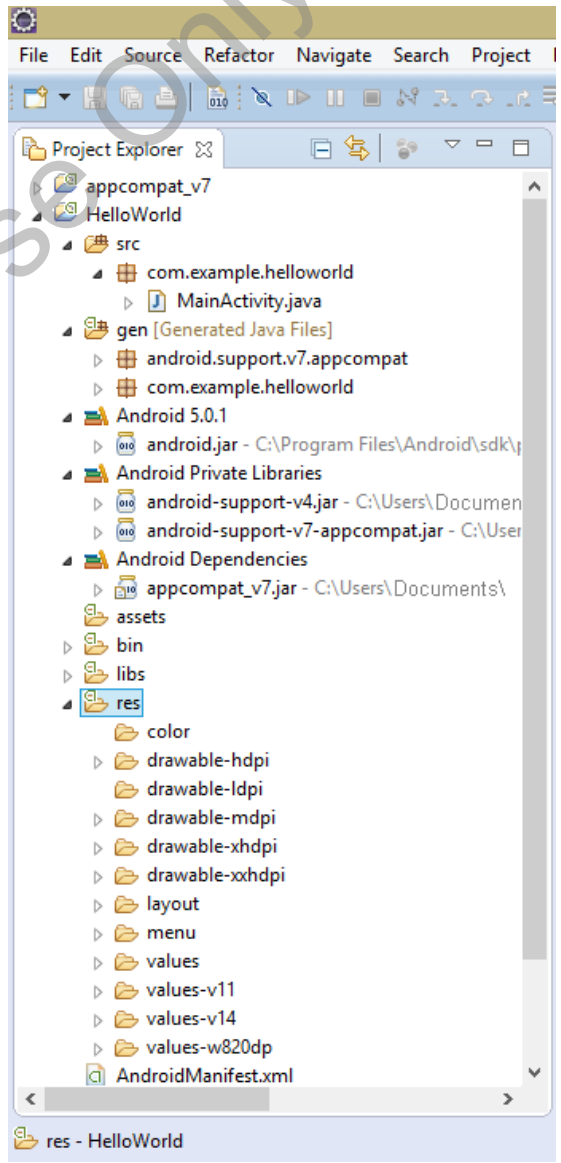
Debugging the Project

- In Eclipse, click Run → Debug.
- The line by line debug reports are displayed in eclipse as shown in the following figure:



Understanding the HelloWorld Project

- **Parts of the Project**
  - src
  - gen
  - assets
  - bin
  - res
  - AndroidManifest.xml
  - project.properties
  - proguard-project.txt
  - ic_Launcher-web.png
  - activity_main.xml
  - .java files



Android Application Fundamentals

- Each Application consists of several components
- An application may include Activities, Services, Content Providers, and so on

- **Security Sandboxes in Android**
  - Every application has a different user ID
  - Permission is granted to the application files so that only valid users can access them
  - The presence of a Virtual Machine (VM) isolates the application

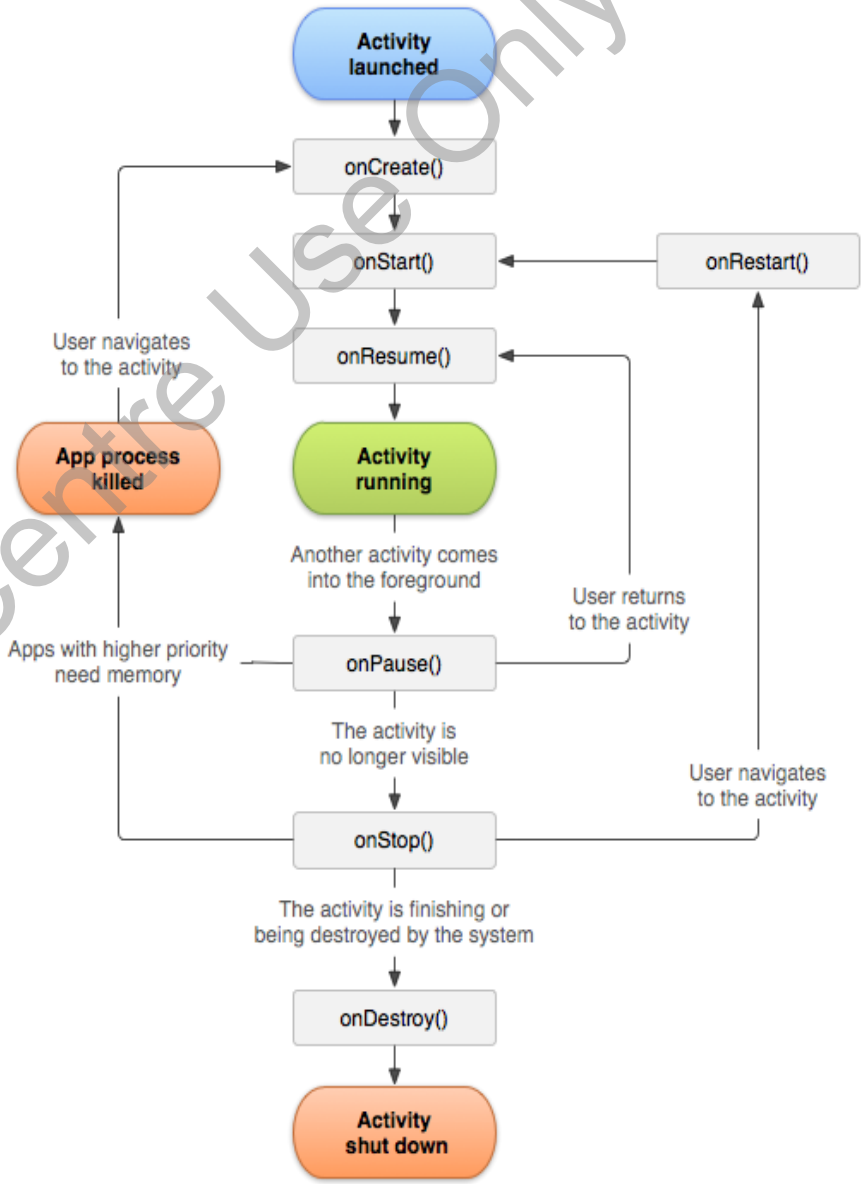
- **Concept of Least Privilege**
  - Two applications can share Linux User ID and VM and access each other's files
  - Permissions granted at time of installation

Application Components

- Activities
- Services
- Content Providers
- Broadcast Receivers
- Resources
- Assets
- Layouts

Activity Life Cycle

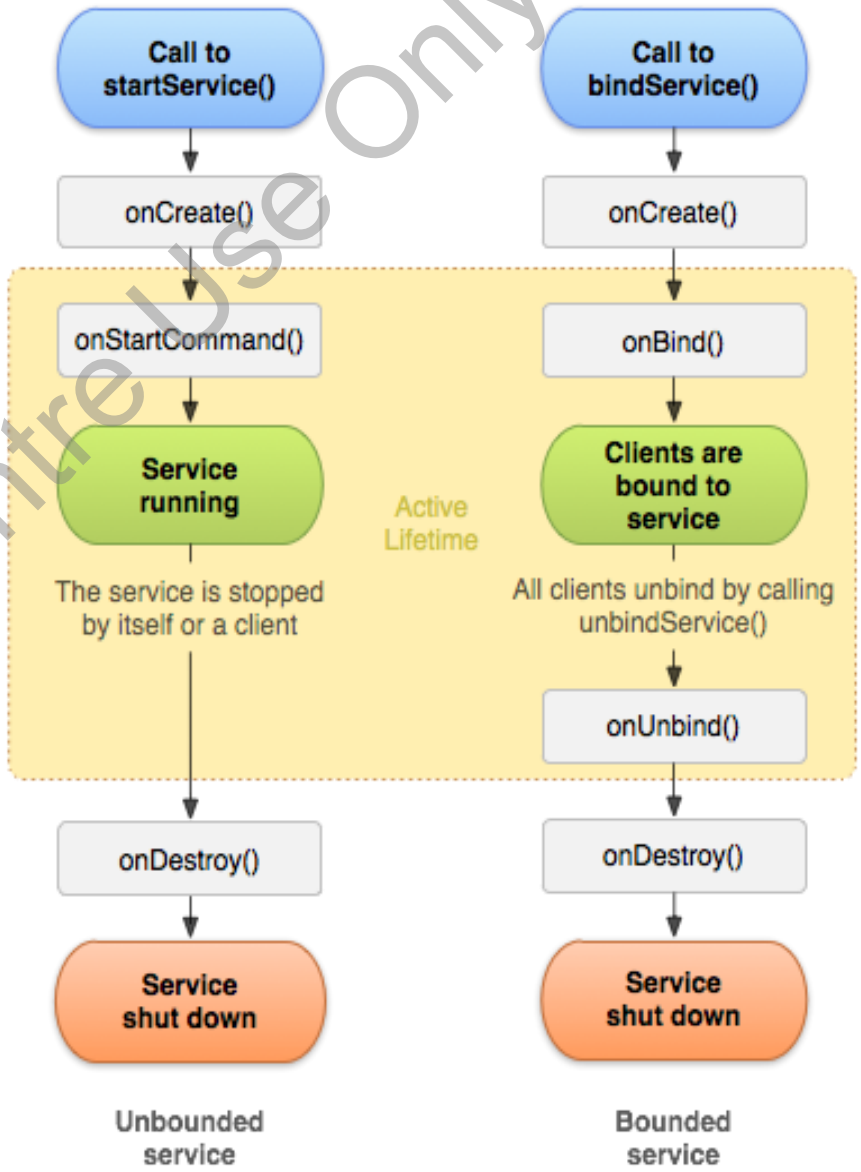
- **States**
  - Active State
  - Paused State
  - Stopped State
  - Inactive State
- **Methods**
  - onCreate()
  - onStart()
  - onResume()
  - onPause()
  - onStop()
  - onDestroy()
  - onRestart()
- **Lifetimes**
  - Full
  - Visible
  - Active



Service Life Cycle

- **Service Forms**
  - Started
  - Bound

- **Methods**
  - Context.startService()
  - onStartCommand()
  - onCreate()
  - Context.stopService() or stopSelf()
  - Context.bindService():
  - onBind()



Content Providers

- **Content Providers**
  - Provide access to data
  - Allow CURD operations
- **Methods**
  - onCreate ()
  - query (Uri, String [], String, String [], String)
  - insert (Uri, ContentValues)
  - update (Uri, ContentValues, String, String [])
  - delete (Uri, ContentValues, String, String [])
  - getType (Uri)

Broadcast Receivers

- **Broadcast Receiver**
  - Receives and responds to announcements
  - Informs applications of events

- **Methods**
  - abortBroadcast()
  - clearAbortBroadcast()
  - getAbortBroadcast()
  - getDebugUnregister()
  - getResultCode()
  - getResultData()

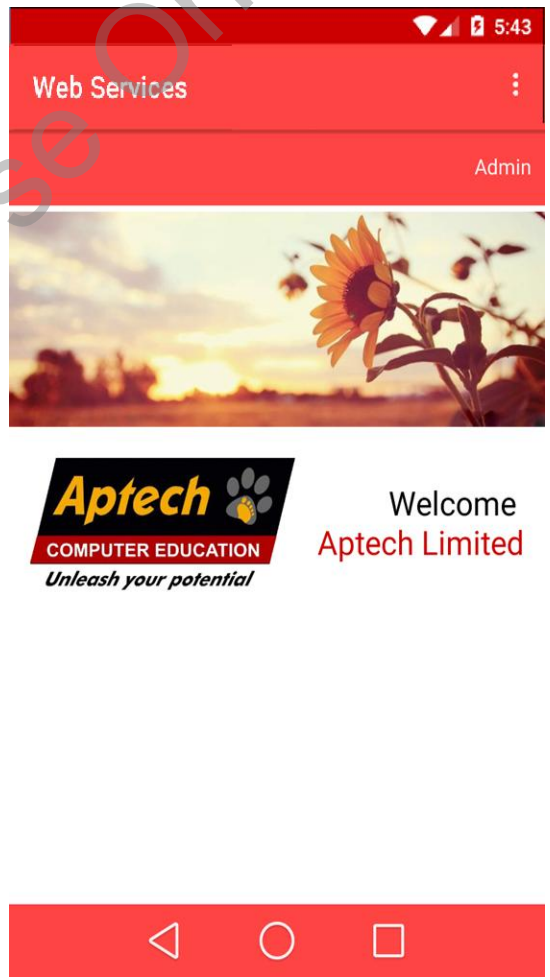
Resources and Assets

- **Resources**
  - Static Content
  - Structured Data
  - Several types of resources
  - Animation, color, drawable, and so on

- **Assets**
  - Functionally similar to resources
  - Non structured data
  - Behaves similar to a file system

Layouts

- **Layouts**
  - Define Visual Structure of the UI
  - Defined in XML
  - Can be edited in code or GUI
- **Types**
  - Linear Layout
  - Relative Layout
  - Table Layout
  - Absolute Layout
  - Frame Layout
  - List View
  - Grid View

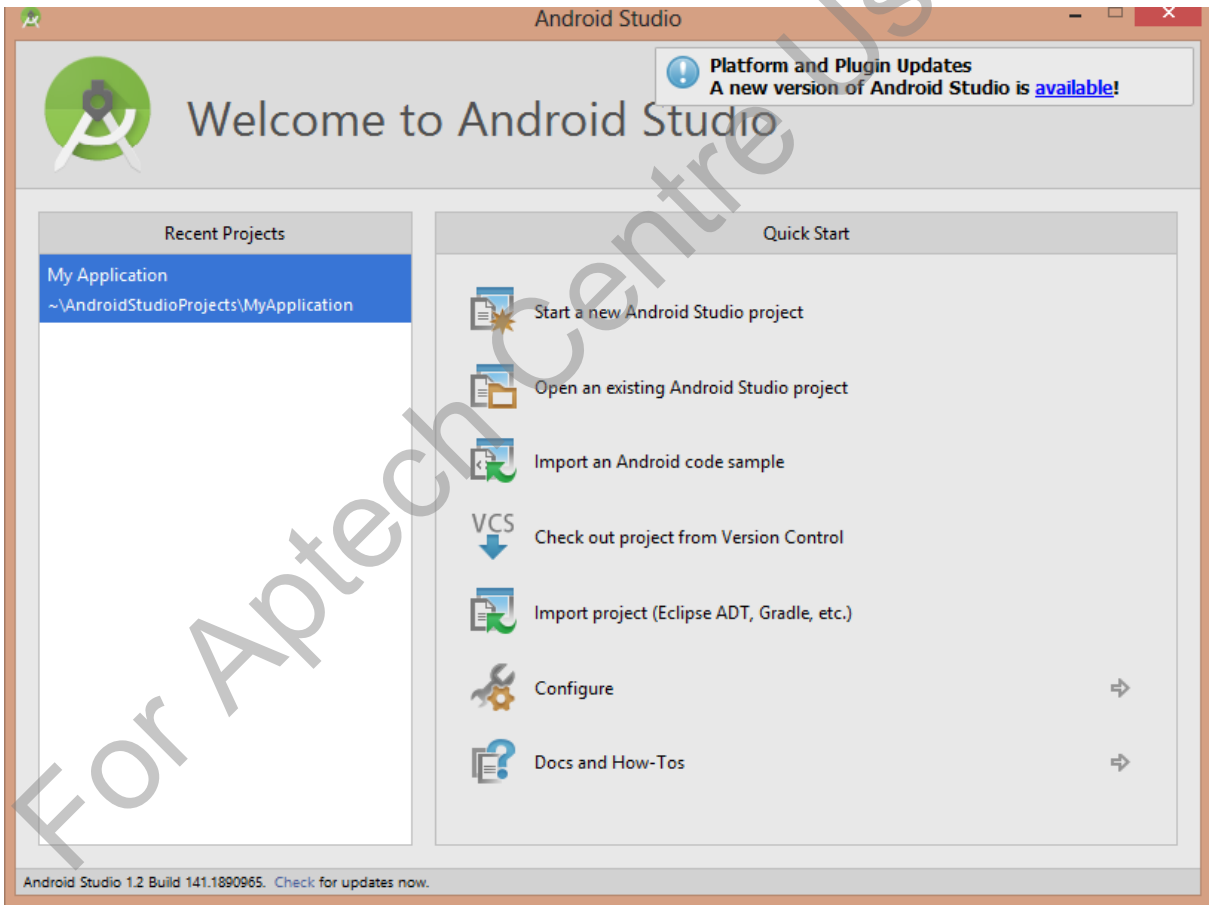


Application Core Concepts

- Intent
  - It Activates Components
- Intent Filter
  - Use to select the Intents that can be managed
- Pre-existing Components
  - Intent Object
  - Intent Matching
  - Intent Resolution
- Android Manifest File
  - Contains Configuration Information
  - Defined in XML
  - Mandatory for all Applications

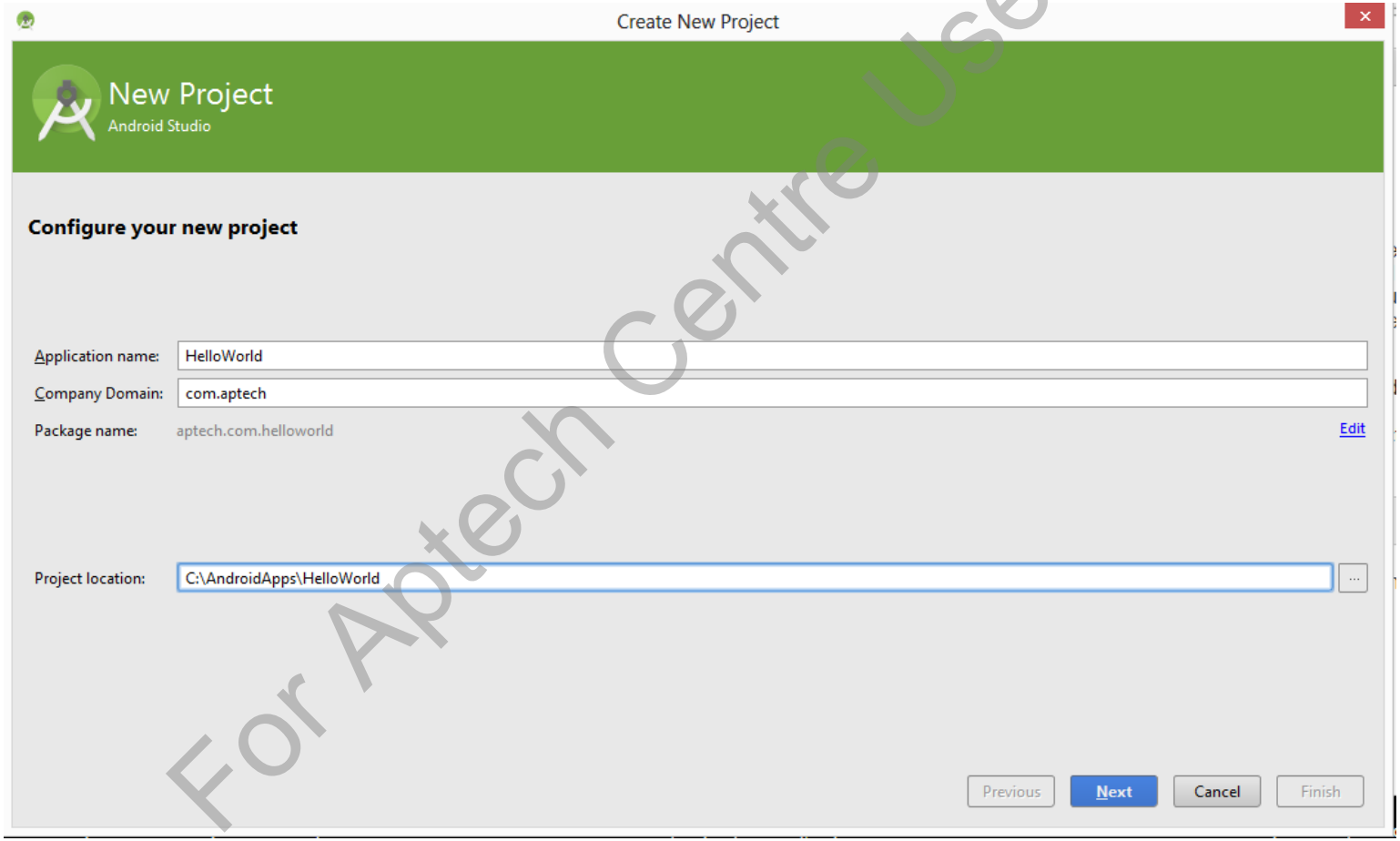
Recreating HelloWorld in Android Studio 1-6

- Start Android Studio
- Select New Android Project from the Welcome screen as shown in the following figure:



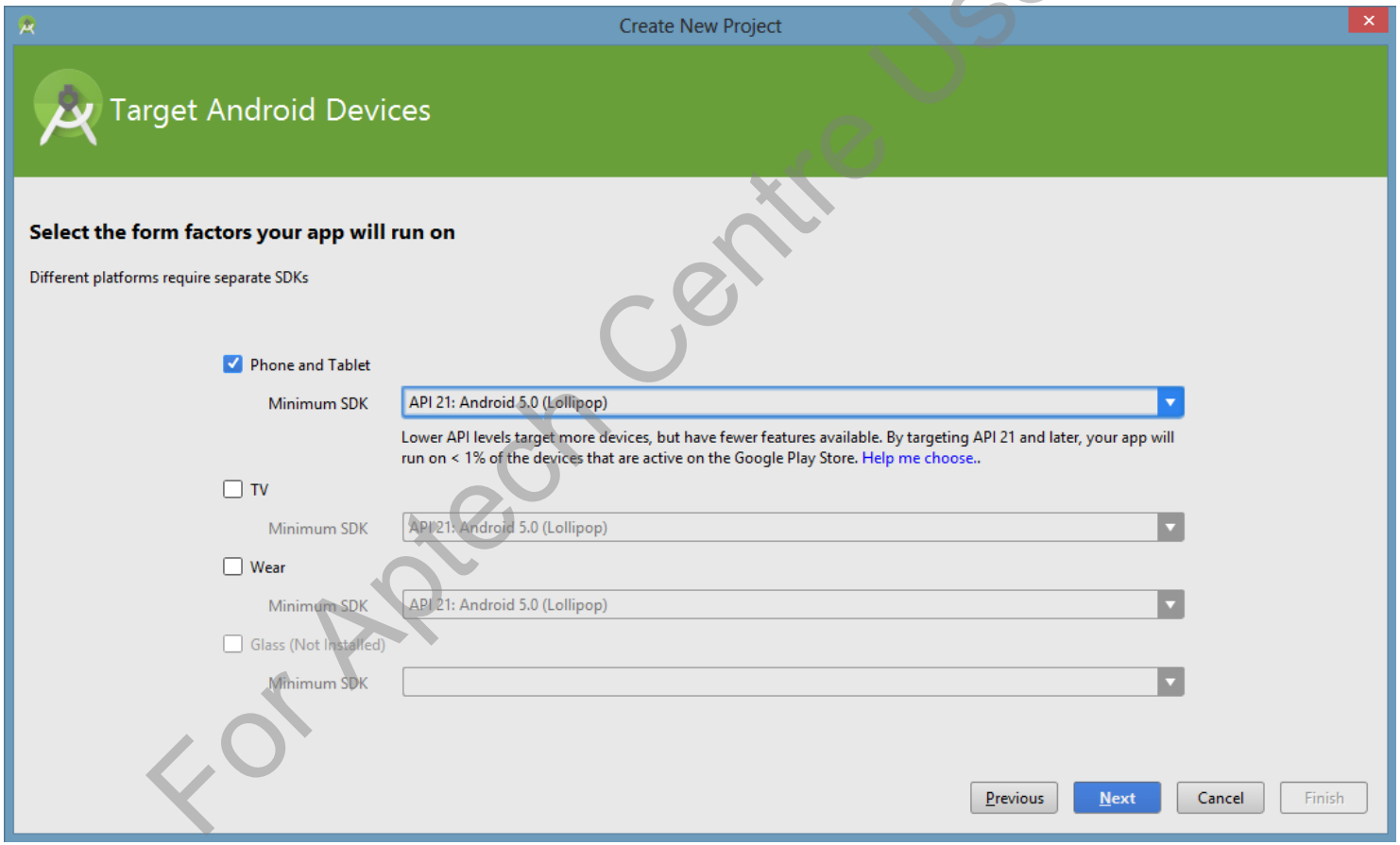
Recreating HelloWorld in Android Studio 2-6

- Enter the details and click Next as shown in the following figure:



Recreating HelloWorld in Android Studio 3-6

- Select API level 21 and click Next as shown in the following figure:

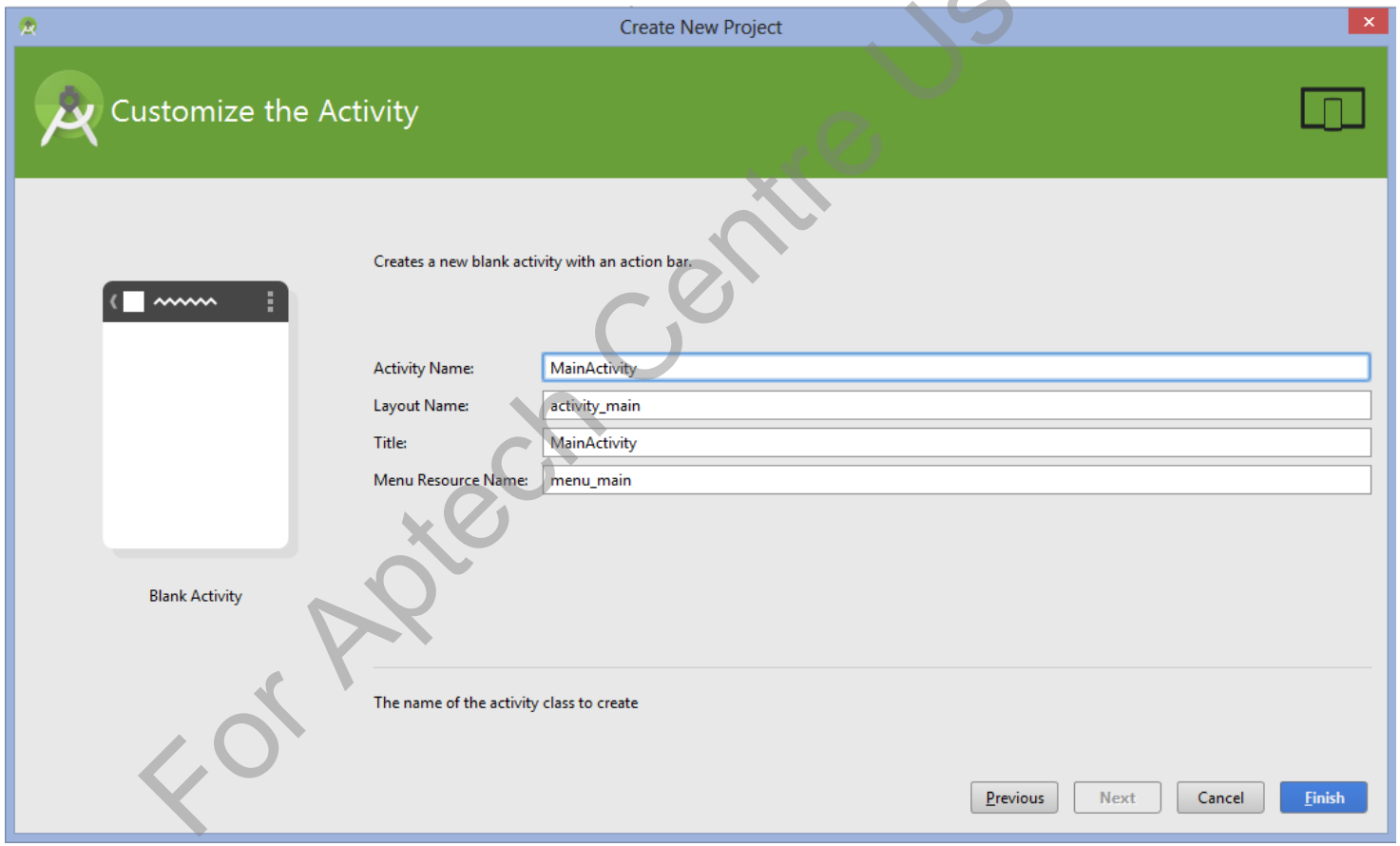


Recreating HelloWorld in Android Studio 4-6

- Select Blank Activity and click Next as shown in the following figure:

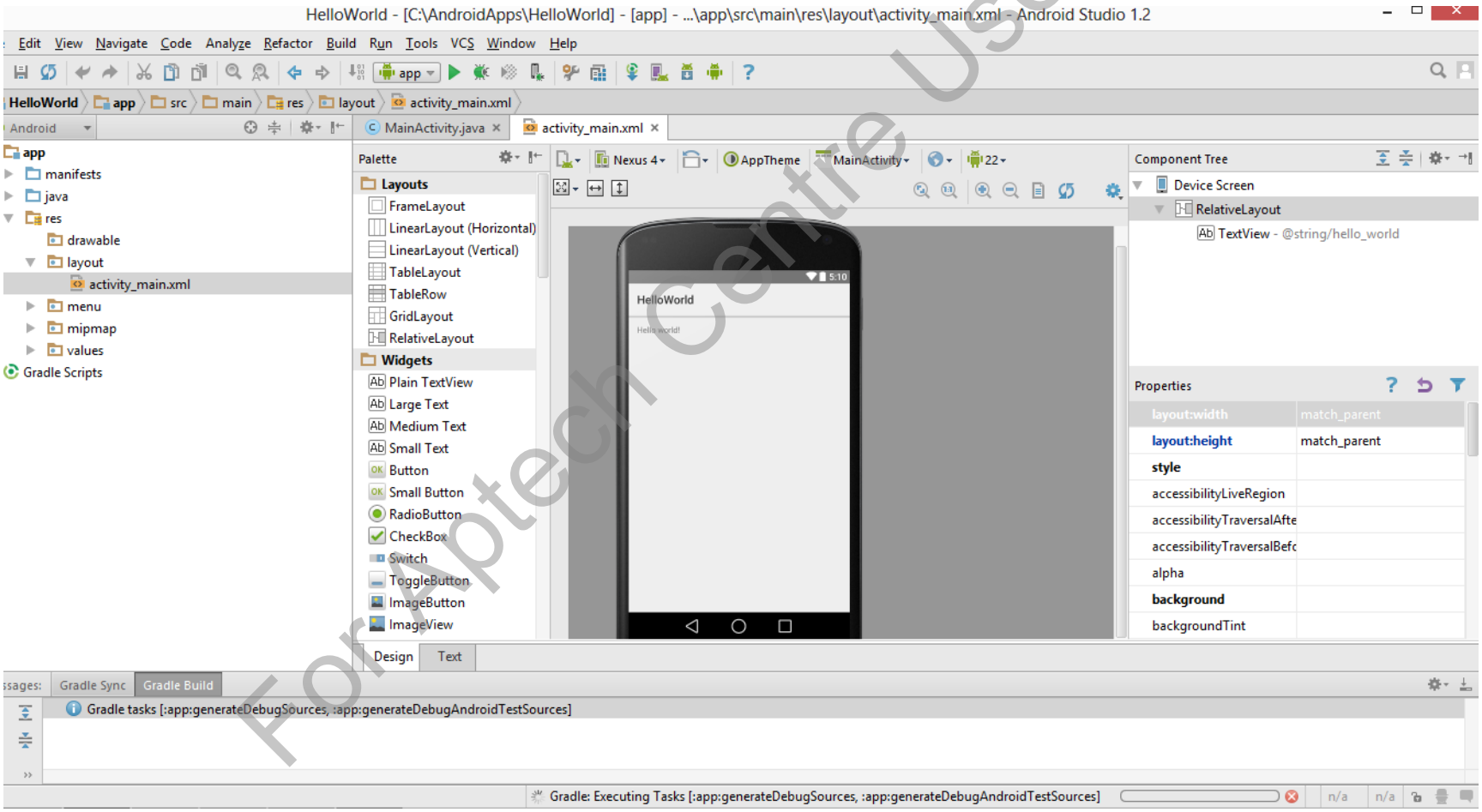
Recreating HelloWorld in Android Studio 5-6

- Enter the Activity details and click Next as shown in the following figure:



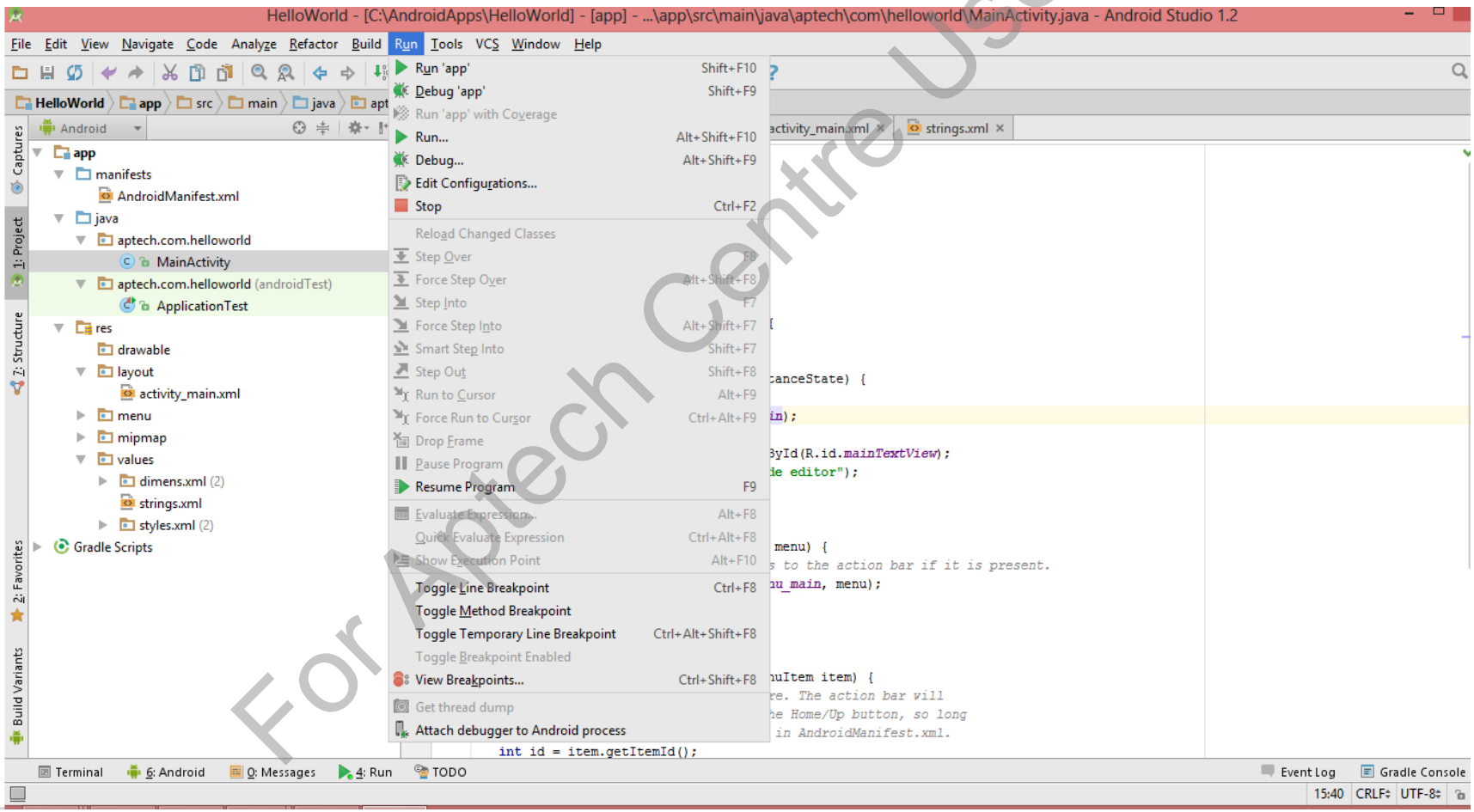
Recreating HelloWorld in Android Studio 6-6

- Click Finish
- The project is created as shown in the following figure:



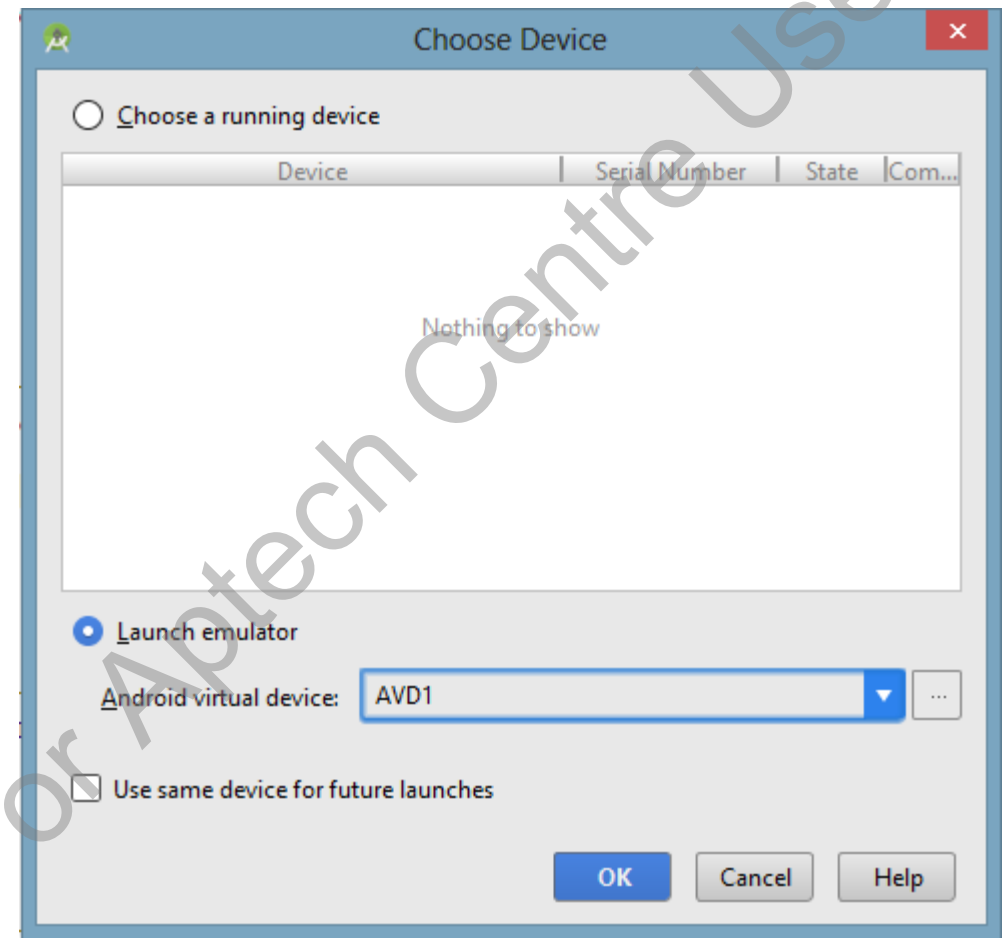
Running the Project 1-3

- Navigate to the ‘run’ menu and click run app as shown in the following figure:



Running the Project 2-3

- From the device selection screen, select AVD1 as shown in the following figure:

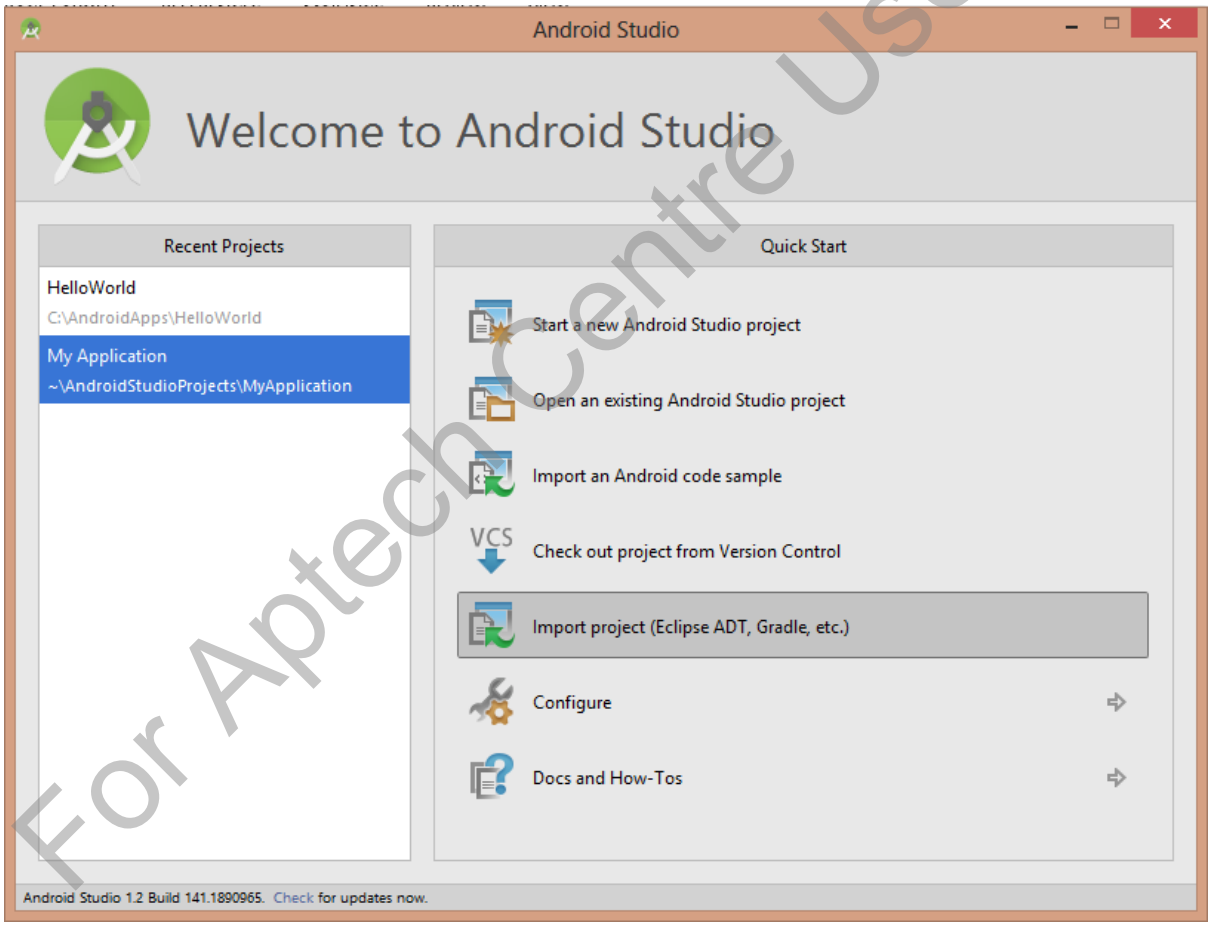


Running the Project 3-3

- The emulator starts with the application running as shown in the following figure:

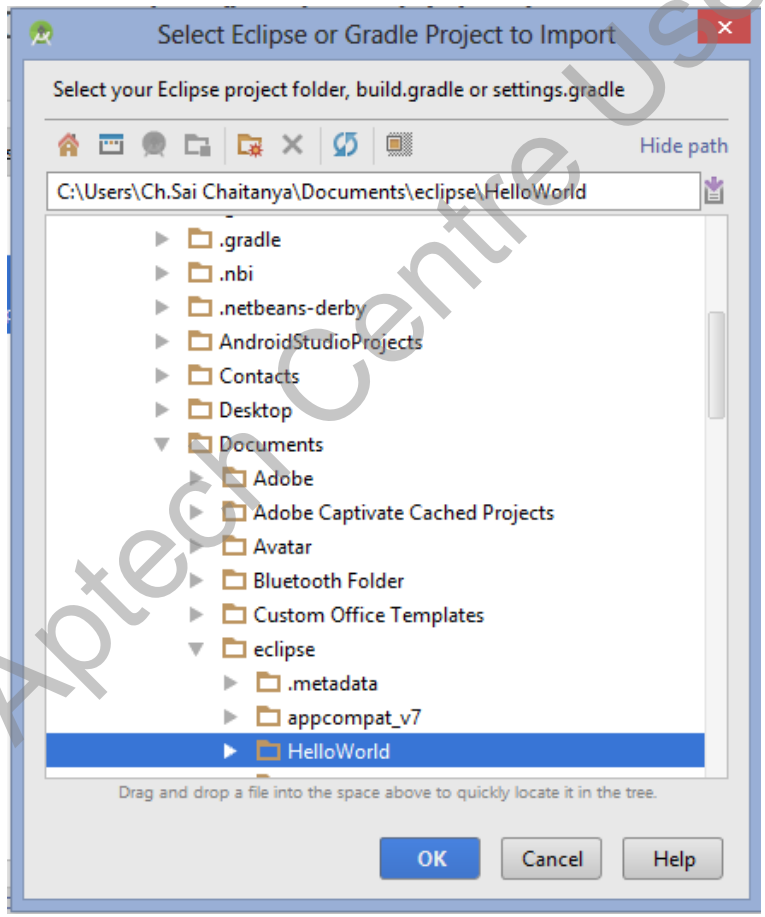
Importing Projects 1-5

- From the Home Screen select Import Project as shown in the following figure:



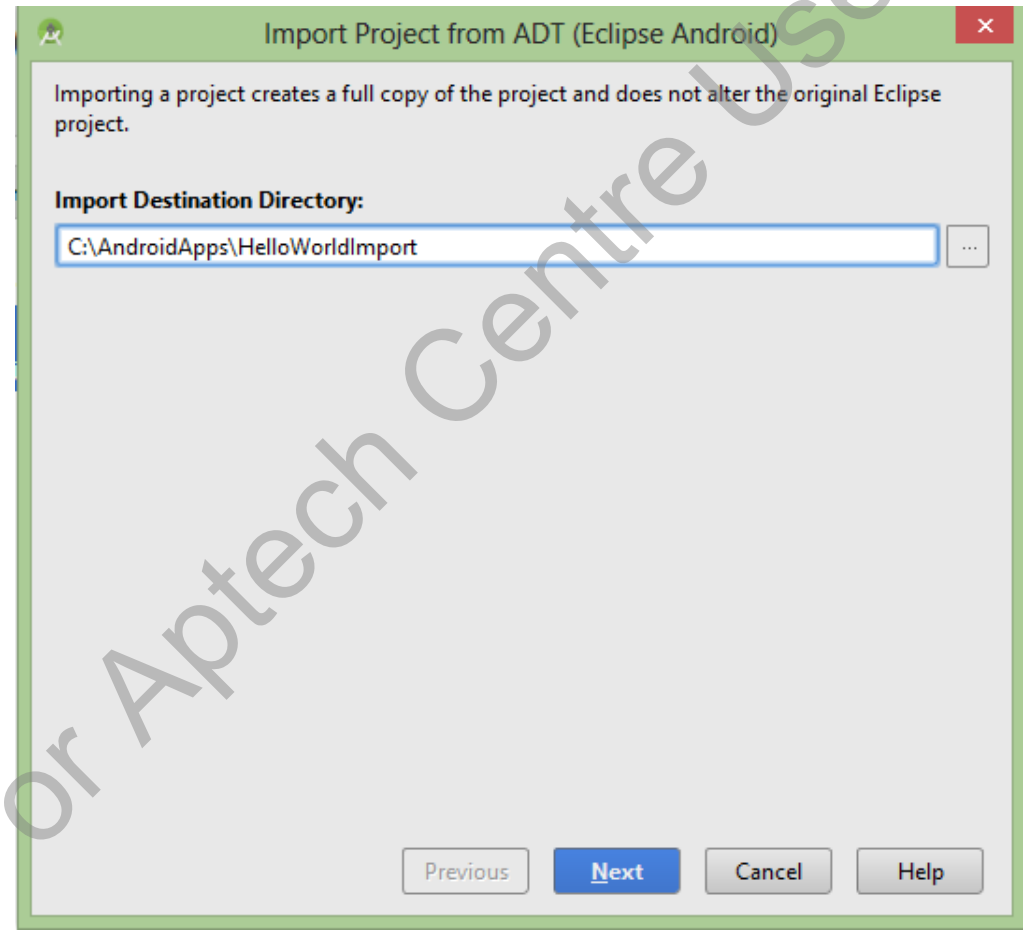
Importing Projects 2-5

- Navigate to the Eclipse project directory and select the folder as shown in the following figure:



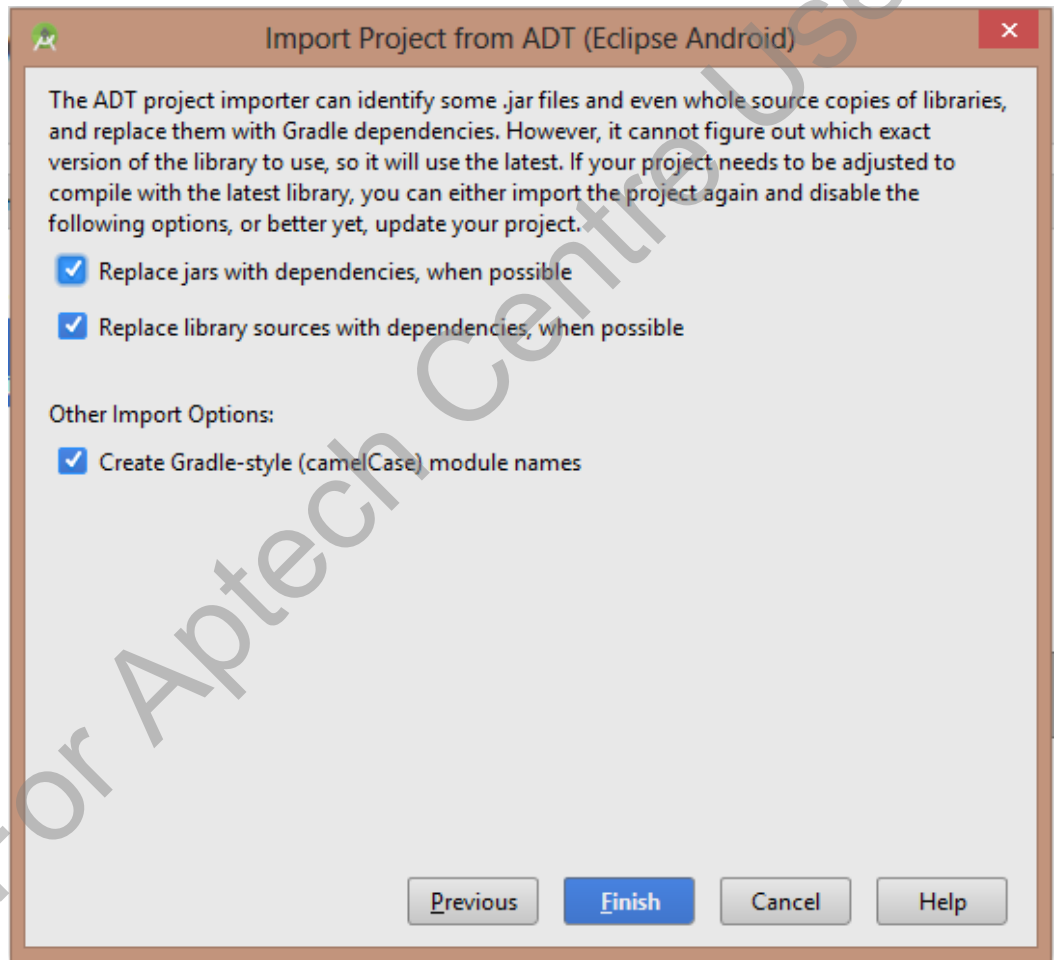
Importing Projects 3-5

- Select the destination directory and click Next as shown in the following figure:



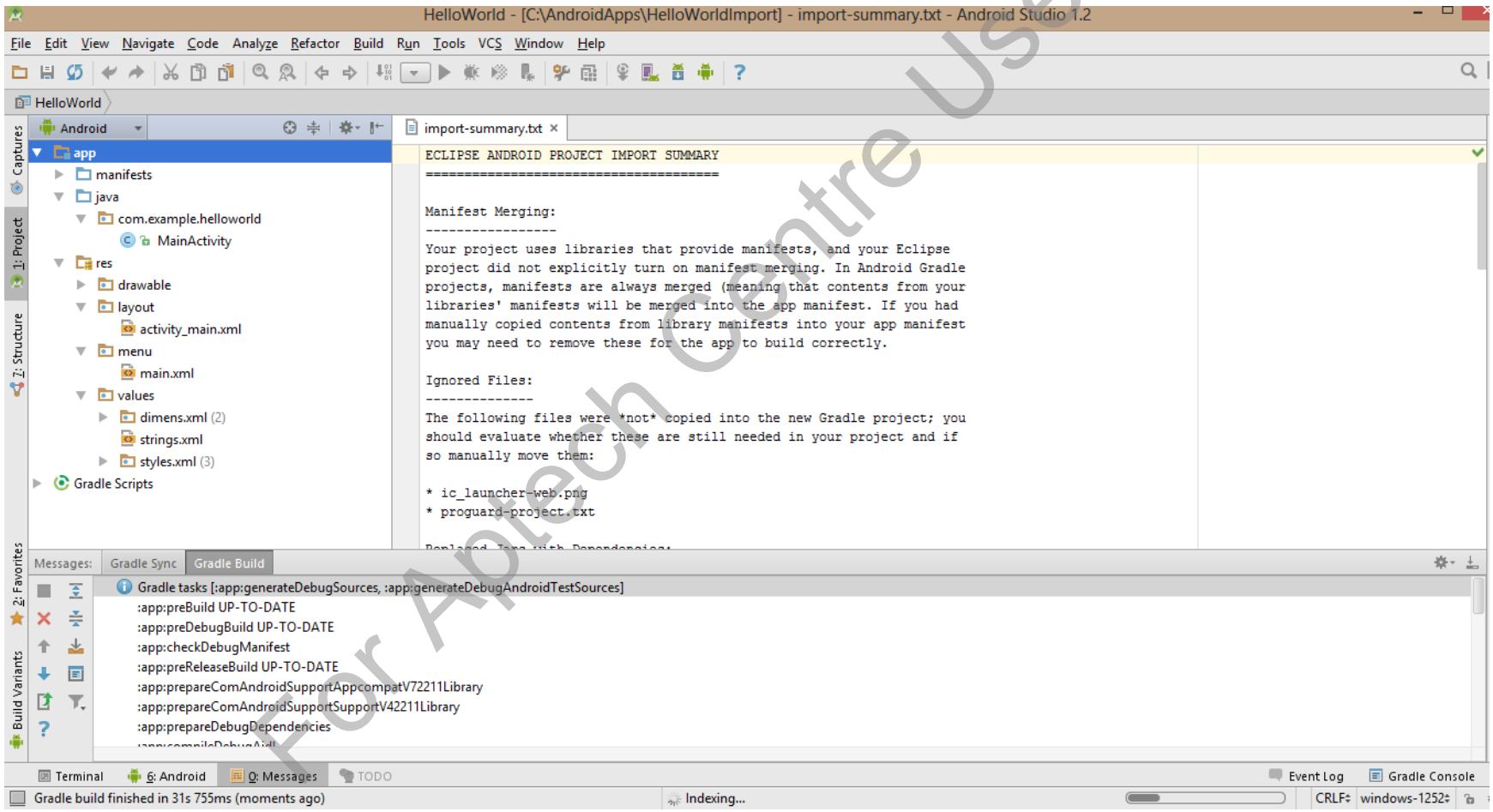
Importing Projects 4-5

- In the confirmation dialog box, click Finish as shown in the following figure:



Importing Projects 5-5

- The project is created and displayed as shown in the following figure:



Deploying Applications on a Real Device 1-2

- **Phone Setup**
  - Navigate to Settings → About Phone
  - Tap on the Build Number Seven Times
  - Return back to the previous screen
  - The Developer Options setting is available
  - Click Developer Options
  - Select the USB Debugging check box

On older versions of Android prior to 4.2, the Developer Options settings is directly available.

Deploying Applications on a Real Device 2-2

- **System Setup**
  - Install the adb drivers for Windows
  - Connect the device to your computer
  - Start command prompt and Navigate to SDK directory → platform-tools
  - Use the command ‘adb devices’
  - The connected Android Device should be listed here
  - Start Android Studio, Open the project, and Navigate to the Run menu
  - Run and Click Run ‘app’
  - From the device selector screen
  - Choose a running Device
  - Select the connected Android Device
  - Click OK

Summary

- An Android project consists of all the files and resources required to build the project into a .apk file for installation
- Activities in Android are UI Screens. Layouts are used to define the UI of these screens
- Services in Android are background processes with no UI
- Content providers are used for managing shared data sets
- Broadcast Receiver receives or responds to announcements broadcast by the system
- Resources and assets are used to hold static content of the application
- Applications can be tested on AVDs or Real Devices