

### UNIVERSITY INSTITUTE OF ENGINEERING

### **Department of Computer Science & Engineering**

## Subject Name: MOBILE APPLICATION DEVELOPMENT LAB Subject Code: 20CSP-356

Submitted to:

Arshdeep sir (E13021)

Submitted by:

Name: Anshuman Singh UID: 20BCS2665 SECTION : 20BCS\_DM\_902 GROUP : A

### **INDEX**

Ex. No	List of Experiments	Conduct	Viva	Record	Total	<b>Remarks/Signature</b>
		(MM: 12)	(MM: 10)	(MM: 8)	(MM: 30)	
1 1	T / 11 <sup>.</sup> 1 ·					
1.1	applications on and running					
1.2						
1.3						
2.1						
2.1						
2.2						
2.2						
2.2						
2.3						
2.4						
3.1						
3.2						
3.3						
5.5						

### **Experiment 1.1**

- 1. Aim: Installing and running applications on android studio.
- 2. Objective: Installing and running applications on android studio.

Android Studio is the official IDE (Integrated Development Environment) for Android app development and it is based on JetBrains' IntelliJ IDEA software. Android Studio provides many excellent features that enhance productivity when building Android apps.

#### 3. System Requirements:

- Microsoft Windows 7/8/10 (32-bit or 64-bit) or Mac os
- 4 GB RAM minimum, 8 GB RAM recommended (plus 1 GB for the Android Emulator)
- 2 GB of available disk space minimum, 4 GB recommended (500 MB for IDE plus 1.5 GB for Android SDK and emulator system image)
- 1280 x 800 minimum screen resolution
- Java JDK5 or later version
- Java Runtime Environment (JRE) 6 Android Studio

#### 4. Steps/Program:

#### **Installation Guide:**

<u>Step 1:</u> Head over to this link <u>https://developer.android.com/studio/#downloads</u> to get the Android Studio executable or zip file.

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

Click on the "I have read and agree with the above terms and conditions" checkbox followed by the download button.



Before installing Android Studio or the standalone SDK tools, you must agree to the following terms and conditions.



Click on the Save file button in the appeared prompt box and the file will start downloading.

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



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.

Click on next.



<u>Step 5:</u> 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'.



# 🧕 Verify Settings

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

Total Download Size: 779 MB		
Sdk Components to Download:		
Android SDK Build-tools, revision 23.0.2	36.3 MB	
Android SDK Platform-tools, revision 23.0.1	2.37 MB	
Android SDK Tools, revision 24.4.1	98 MB	
Android Support Repository, revision 25	149 MB	
Google APIs Intel x86 Atom System Image, Google Inc. API 23, revisio	n 9 333 MB	
Google APIs, Android API 23, revision 1	176 KB	
Google Repository, revision 23	62.1 MB	
Intel x86 Emulator Accelerator (HAXM installer), revision 5.5.0	219 KB	
SDK Platform Android 6.0, API 23, revision 2	67.1 MB	
Sources for Android SDK, API 23, revision 1	30.3 MB	

Click the **OK** button.

<u>Step 6:</u> 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	N

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



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 Dracula. Choose as required.

Click on the Next button.

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

	HAXM installation wants to make changes. Type your password to allow this.			
	Username: Ben Deitch			
	Password: PASSWORD			
	Cancel OK			
	Android Studio Setup Wizard			
Þ	ownloading Components			
Installed Android Support Repository, revision 25 Installing Google Repository, revision 23 Installing Intel x86 Emulator Accelerator (HAXM installer), revision 5.5.0 Installing Intel x86 Emulator Accelerator (HAXM installer), revision 5.5.0 Installed Intel x86 Emulator Accelerator (HAXM installer), revision 5.5.0 Installing Android SDK Platform-tools, revision 23.0.1 Stopping ADB server failed (code -1). Installed Android SDK Platform-tools, revision 23.0.1 Installed Sources for Android SDK, API 23, revision 1 Installed Sources for Android SDK, API 23, revision 1 Installed Sources for Android SDK, API 23, revision 1 Installed Google APIS Intel x86 Atom System Image, Google Inc. API 23, revision 9 Installed Google APIS Intel x86 Atom System Image, Google Inc. API 23, revision 9 Installed Android SDK Tools, revision 24.4.1 Updated ADB too Support the USB devices declared in the SDK add-ons. Stopping ADB server succeeded. Starting ADB server succeeded. Done. 10 packages installed. Android SDK is up to date. Running Intele HAXM installer Silent installation Pass! Creating Android virtual device Android virtual device Nexus_5_API_23_x86 was successfully created Cancel Previous Next Finish				

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

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.



#### **Running Applications on Android Studio:**

The panel on the left side of the android studio window has all the files that the app includes. Under the java folder, observe the first folder containing the java file of your project.



For every activity, a ".java" file and a ".xml" file is created. In this case for MainActivity, "MainActivity.java" and "activity\_main.xml" are created.

The above java file shows us the default code that is present when an app is created. An activity is created that extends AppCompactActivity class.

The "res" folder contains "layout" subfolder, which includes the xml files of the projects.



You can find the activity\_main.xml file under the layout folder. This the XML file corresponding to the MainActivity. There is an onCreate function that overrides a function of AppCompactActivity class. onCreate(Bundle) is where you initialize your activity. When the activity is first started, then both onCreate() methods are called. But after the first start of Activity, the onCreate() of application will not be called for subsequent runs.

• Now, consider the activity\_main.xml file, it contains various tags similar to HTML. The first tag ensures the version. The second tag is usually the Layout tag. There are various types of Layouts but for now, let us go with the default RelativeLayout. This is a layout that places the widgets relative to screen size.

There is a TextView widget by default. This "TextView" is basically the Text field that displays the text specified. It has various attributes. For now, consider the default attributes present. The layout\_width and layout\_height are the width and height of the widget occupied in the screen. The attribute "wrap\_content" refers to width or height being restricted to the content of the text.

The text attribute takes a string in quotations ( i.e., " " ). The content within this is displayed on the screen.



Now, click the "Run" option at the Toolbar at the top. You can observe the option being highlighted in the image below.

You would get a pop-up as in the image below.

My Application - [~/AndroidStudioProjects/MyApplication2] - [app] - ~/AndroidStudioProjects/MyApplication2/app/src/main/res/layout/act 🛜 En 🖇 💌 💷 (22%) 🍬 9:5	7РМ 🖞 Апоор
Eile Edit View Navigate Code Analyze Befact(Run'app'(Shift+F10) VCS Window Help	
Q □ H Ø ダ ル ル ⑥ ◎ Q Q ← → H ⊕ app~ ≥ ≥ ● ● ◎ ↓ G ■ 9 ⊕ ♀ E ≛ ♣ ?	Q, 📄
🚬 MyApplication2) 🖻 app ) 🗖 src ) 🗖 main ) 🖬 res ) 🖻 layout ) 🗃 activity main.xml )	
🔨 🛃 🐥 Android 🔹 🕫 🛊 tr 📓 <u>activity main.xml</u> × 📧 MainActivity.java ×	۲
line to the second sec	Gra
S A D manifests	9 <del>6</del>
V java v java v com.example.anoop.myapplicatic v com.example.anoop.m	
Figure 4 and 50 (1 kyour hatting in a constraint) and 50 (1 kyour hatting in and 50 (1 kyour hattin	evie
android:paddingLet="iddp"	<
android:padding10p+"16dp"	
D drawable tools:context="con.example.anoop.myapplication.HainActivity">	
See	
android: layout, height "wrap.content"	
► Improp A android:text=Mello World!" /> C-RelativeLown+	
▶ ⓒ Gradle Scripts	
	÷
	Andr
	oid
	ó
Design Text	<u>1</u>
🔤 Terminal 🔶 & Android Monitor 📄 & Messages 🔮 TODO 👘 Event Log 🗑 G	radle Console

You can either choose the emulator or you can connect your phone and find them listed under Connected Devices but for this you must enable the developer options in your phone and set the USB debugging mode on.

My Application - [~/AndroidStudioProjects/MyApplicat	ion] - MainActivity.java - Android Studio 2.1.2	奈 🖪 🕴 🖂 🛄 (17%) 🐗 10:06 PM 🔱 Anoop
Eile Edit View Navigate Code Analyze B	factor <u>B</u> uild R <u>u</u> n <u>T</u> ools VC <u>S</u> <u>W</u> indow <u>H</u> elp	
Q □ H Ø ≠ ≯ X D 0 Q Q 4 →	48 🗰 app 💌 🕨 🛎 🛞 🕵 G 🔳 🛠 🏦 😫 💁 暮 🗧	Q, 🗖
MyApplication2 app 5 src and	📬 res 🕽 🛅 layout 🕽 🖻 activity, main.xml 🛇	
	Select Deployment Target	
S S S S S S S S S S S S S S S S S S S	No USB devices or running emulators detected Troublest	
V Dijava	Copported Devices	
v 🖸 com.example.anoop.myapplicat		
>_ B com example appen myapplicat	Available Emulators	
Com example apoop myapplicat	Nexus Marshmallow mdpi	
🗖 drawable		
👘 🖉 🔻 🗖 layout		
activity_main.xml		
👸 🕨 🖻 mipmap		
Values		
tes internet		
E Favor	Create New Emulator	
	Use same selection for future launches	ncel
ariants		Anor
	Design Text	ž
🗷 📧 Terminal 🛛 🐥 <u>6</u> : Android Monitor 🛛 🗮 <u>0</u> : Me	ssages 🔮 TODO	🗮 Event Log 🛛 🖃 Gradle Console
🔲 🔲 Run selected configuration		2:1 LF: UTF-8 Context: <no context=""> 🔉 🚇</no>

Once done, click on OK.

Usually, the emulator consumes a lot of RAM. The more RAM size you have, the faster your emulator will work. Generally, 4GB is the descent RAM size. Size more than that would increase the performance of your emulator.

The image below shows the working of the first app, My Application. You can find all the basic functionalities that your phone has, on the emulator, like Home button, back button, power, etc.



#### Learning outcomes (What I have learnt):

- Learnt about installation of android studio.
- Learnt about running application on android studio.

#### Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			