SCHEME OF VALUATION

Revision 2015 Course Title : SMART DEVICE PROGRAMMING			Course Code 6133	
Q.No	Scoring Indicator	Split up score	Sub Total	Total
I	**PART A**			
1	Launcher icons reside in the mipmap. This folder contains the ic_launcher.png image that appears when you run the default app	2	2	
2	It is the "glue" that enables different activities from different applications to work together. When your application has more than one activity, you may need to navigate from one activity to another. In Android, you navigate between activities through what is known as an intent.	2	2	10
3	<ImageView android:src = "@drawable/img" android:layout_width="wrap_content" android:layout_height="wrap_content" />	1 1	2	
4.	WRITE_EXTERNAL_STORAGE	2	2	
5	**Any two attributes** **onabort** —Fires when an action is aborted. **onbeforeonload , onbeforeonunload , and onunload** —Fires just before an element loads or unloads and as an element unloads. . **oncontextmenu** —Fires when the context menu is triggered. . **ondrag , ondragend , ondragenter , ondragleave , ondragstart , and ondrop** —These fire when various drag-and-drop actions occur. . **onerrror and onmessage** —These fire when errors or messages are triggered. . **onscroll** —This fires when the user scrolls the browser scroll bar. . **onresize** —Fires when an element is resized.	2*1	2	
II	**PART B**			
1	To download Android Studio for you computer, you need to visit official site of android. After downloading android studio, follow below steps to install android studio on your computer. Step1: Open you downloaded android studio file and wait for few seconds, it will take some time to load on your screen. Step2: After that the window will appear on your screen, where it will show you welcome to android studio. Go for "Next" Step3: Now choose what you want to install, all are most important component in android programming. Go for "Next" Step4: Go for "I Agree", It's for integrating SDK, it will confirm SDK installation. Step5: Go for "I Agree", It's for Haxm installation. Step6: Now choose the location for Android Studio and SDK. Go for "Next"	3*2	6	

	Step7: Here you need to define the size of android emulator processor. You can define up to 2GB and more than that, depends on your RAM capacity. The size will be determined through Haxm. This will help your android studio to run faster. Step8: Now choose setting for shortcut of android studio on your computer. Step9: After that it will take lots of time to copy all the files to the storage location you have selected before.			
2	Any 4 features 1. Gradle-based build support. 2. Android-specific refactoring and quick fixes 3. Lint tools to catch performance, usability, version compatibility and other problems 4. ProGuard and app-signing capabilities 5. Template-based wizards to create common Android designs and components. 6. **A rich layout editor:** it allows you to drag-and-drop UI components, preview layouts on multiple screen configurations. Preview appears instantly as you change in the layout editor. You can choose a language, and can see the preview of layout with that locale. 7. **Rich Color Preview editor:** While adding colors as a resource, and we can see the color preview at the left hand side of the editor. 8. **Deep Code Analysis:** If you point to a line and it gives detailed explanation about an exception based on the annotation added. And you can also know which constants are allowed for which API. It also has the powerful code completion. You can also inspect code in whole project, InteliJ lists all Lint errors during code inspection.	4 * 1.5	6	

3	In your current Activity, create a new Intent:
	String value="Hello world";
	Intent i = new Intent(CurrentActivity.this, NewActivity.class);
	i.putExtra("key",value);
	startActivity(i);
	Then in the new Activity, retrieve those values:
	Bundle extras = getIntent().getExtras();
		3	6	

	if (extras != null) {
	String value = extras.getString("key");
	//The key argument here must match that used in the other activity
	}
		3		

4	To send an SMS message programmatically, you use the SmsManager class. You call the getDefault() static method to obtain a SmsManager object. You then send the SMS message using the sendTextMessage() method:
	private void sendSMS(String phoneNumber, String message)
	{
	SmsManager sms = SmsManager.getDefault();
	sms.sendTextMessage(phoneNumber, null, message, null, null);
	}
	
	In the AndroidManifest.xml file, add the following statements
	<uses-permission
	android:name="android.permission.SEND_SMS"></uses-permission>
		4
		
		2
			6	

5	public class MainActivity extends ListActivity {
	String[] days = {
	"Sunday","Monday","Tuesday","Wednesday","Thursday","Friday","Saturday"
	};
	@Override
	public void onCreate(Bundle savedInstanceState) {
	super.onCreate(savedInstanceState);
	setListAdapter(new ArrayAdapter<String>(this,
	android.R.layout.simple_list_item_1, presidents));
	}
	public void onListItemClick(
	ListView parent, View v, int position, long id)
	{
	Toast.makeText(this,
	"You have selected " + days[position],
	Toast.LENGTH_SHORT).show();
	}
	}
		1
		
		1
		
		2
		
		2
			6	

6	<audio> —Embedded sound files. <canvas> —Embedded dynamic graphics. <embed> —To add other technologies that don't have a specific HTML5 element. <source> —The source files for embedded sound and video. <track> —Supplementary media tracks for embedded sound and video. <video> —Embedded video files	6*1	6	

7	Steps are
	1. Create your web page as you normally would:
	<!DOCTYPE html>
	<html>
	<head>
	<title>Simple JavaScript</title>
	</head>
	<body>
	</body>
	</html>
	2. Add a script tag to the <head> of the page:
	<script>
	</script>
	3. Add the JavaScript function inside the script tags to write "Hello World" in an
	alert window:
	function hello() {
	alert("Hello World");
	}
	4. Add a link to the body of your document to call the script:
	Click Me
	5. Include a <noscript> tag below the link with alternative text:
	<noscript>
	<p>Hello World
	<p>This text is not written with JavaScript.
	</noscript>
		6		6

III	**Architecture diagram (Refer fig1)** **Explanation** The Android OS is roughly divided into five sections in four main layers: ➤➤ **Linux kernel** — This is the kernel on which Android is based. This layer contains all the low level device drivers for the various hardware components of an Android device. ➤➤ **Libraries** — These contain all the code that provides the main features of an Android OS. For example, the SQLite library provides database support so that an application can use it for data storage. The WebKit library provides functionalities for web browsing. ➤➤ **Android runtime** — At the same layer as the libraries, the Android runtime provides a set of core libraries that enable developers to write Android apps using the Java programming	7 8	15	

	language. The Android runtime also includes the Dalvik virtual machine, which enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine (Android applications are compiled into the Dalvik executables). Dalvik is a specialized virtual machine designed specifically for Android and optimized for battery-powered mobile devices with limited memory and CPU. ➤➤ **Application framework** — Exposes the various capabilities of the Android OS to application developers so that they can make use of them in their applications. ➤➤ **Applications** — At this top layer, you will find applications that ship with the Android device (such as Phone, Contacts, Browser, etc.), as well as applications that you download and install from the Android Market. Any applications that you write are located at this layer.			
IV(a)	**Refer Table1** **Any six points, 1.5 marks each**	6* 1.5	9	
IV(b)	explain any three features of 2G explain any three features of 3G **2G** • Digital modulation was first introduced • Uplink band:890–915 MHz & Downlink band: 935–960 MHz • Multiple Access Scheme: FDMA, TDMA(used in GSM) &CDMA • Data rate: upto 9.6 Kbps • New thing introduced was **digitally encrypted text messages** • Voice calls are free of noise due to digital modulation • Examples are GSM & CDMA **3G** • works on 2100 MHz band • it increased the efficiency of spectrum by compressing audio • it supports upto 2 Mbps speed for stationary or low mobility users and upto 384 Kbps for mobile users. • carrier is at high frequency, hence require more transmitting power • Applications: web browsing, high security, international roaming	3*1 3*1	6	
V	Diagram (Refer diagram 2 given) Explanation The Activity class defines the following events:	8 7		

	➤ ➤ onCreate() — Called when the activity is first created ➤ ➤ onStart() — Called when the activity becomes visible to the user ➤ ➤ onResume() — Called when the activity starts interacting with the user ➤ ➤ onPause() — Called when the current activity is being paused and the previous activity is being resumed ➤ ➤ onStop() — Called when the activity is no longer visible to the user ➤ ➤ onDestroy() — Called before the activity is destroyed by the system (either manually or by the system to conserve memory) ➤ ➤ onRestart() — Called when the activity has been stopped and is restarting again		15	

VI(a)	From your FirstActivity call the SecondActivity using startActivityForResult() method
	For example:
	Intent i = new Intent(this, SecondActivity.class);
	startActivityForResult(i, 1);
	In your SecondActivity set the data which you want to return back to FirstActivity. If you don't want to return back, don't set any.
	For example: In SecondActivity if you want to send back data:
	Intent returnIntent = new Intent();
	returnIntent.putExtra("result",result);
	setResult(Activity.RESULT_OK,returnIntent);
	finish();
	Now in your FirstActivity class write following code for the onActivityResult() method.
	@Override
	protected void onActivityResult(int requestCode, int resultCode, Intent data) {
	if (requestCode == 1) {
	if(resultCode == Activity.RESULT_OK){
	String result=data.getStringExtra("result");
	}
	}
	}//onActivityResult
		3
		3
		3
			9	

VI(b)	These are the three different types of services: **Foreground** A foreground service performs some operation that is noticeable to the user. For example, an audio app would use a foreground service to play an audio track. Foreground services must display a Notification. Foreground services continue running even when the user isn't interacting with the app. **Background** A background service performs an operation that isn't directly noticed by the user. For example, if an app used a service to compact its storage, that would usually be a background service. **Bound** A service is *bound* when an application component binds to it by calling bindService(). A bound service offers a client-server interface that allows components to interact with the service, send requests, receive results, and even do so across processes with interprocess communication (IPC). A bound service runs only as long as another application component is bound to it. Multiple components can bind to the service at once, but when all of them unbind, the service is destroyed.	3*2	6	
VII	Any five layouts Android supports the following ViewGroups: ➤➤ LinearLayout ➤➤ AbsoluteLayout ➤➤ TableLayout ➤➤ RelativeLayout ➤➤ FrameLayout ➤➤ ScrollView **LinearLayout** The LinearLayout arranges views in a single column or a single row. Child views can be arranged either vertically or horizontally. To see how LinearLayout works, consider the following elements typically contained in the main.xml file: <?xml version="1.0" encoding="utf-8"?> <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android" android:orientation="vertical" android:layout_width="fill_parent"	5*3	15	

```
  android:layout_height="fill_parent"
  >
<TextView
  android:layout_width="fill_parent"
  android:layout_height="wrap_content"
  android:text="@string/hello"
  />
</LinearLayout>
```

AbsoluteLayout

The AbsoluteLayout enables you to specify the exact location of its children. Consider the following
UI defined in main.xml :

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout
  android:layout_width="fill_parent"
  android:layout_height="fill_parent"
  xmlns:android="http://schemas.android.com/apk/res/android"
  >
<Button
  android:layout_width="188dp"
  android:layout_height="wrap_content"
  android:text="Button"
  android:layout_x="126px"
  android:layout_y="361px"
  />
<Button
  android:layout_width="113dp"
  android:layout_height="wrap_content"
  android:text="Button"
  android:layout_x="12px"
  android:layout_y="361px"
  />
</AbsoluteLayout>
```

TableLayout

The TableLayout groups views into rows and columns. You use the <TableRow> element to designate a row in the table. Each row can contain one or more views. Each view you place within a row forms a cell. The width of each column is determined by the largest width of each cell in that column.

```
<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_height="fill_parent"
android:layout_width="fill_parent"
>
<TableRow>
<TextView
android:text="User Name:"
```

	android:width ="120px" /> <EditText android:id="@+id/txtUserName" android:width="200px" /> </TableRow> </TableLayout> **RelativeLayout** The RelativeLayout enables you to specify how child views are positioned relative to each other. <?xml version="1.0" encoding="utf-8"?> <RelativeLayout android:id="@+id/RLayout" android:layout_width="fill_parent" android:layout_height="fill_parent" xmlns:android="http://schemas.android.com/apk/res/android" > <TextView android:id="@+id/lblComments" android:layout_width="wrap_content" android:layout_height="wrap_content" android:text="Comments" **android:layout_alignParentTop="true"** **android:layout_alignParentLeft="true"** /> <Button android:id="@+id/btnCancel" android:layout_width="124px" android:layout_height="wrap_content" android:text="Cancel" **android:layout_below="@+id/txtComments"** **android:layout_alignLeft="@+id/txtComments"** /> </RelativeLayout> **Scrollview** A ScrollView is a special type of FrameLayout in that it enables users to scroll through a list of views that occupy more space than the physical display. The ScrollView can contain only one child view or ViewGroup, which normally is a LinearLayout. **Framelayout** The FrameLayout is a placeholder on screen that you can use to display a single view. Views that you add to a FrameLayout are always anchored to the top left of the layout.			
VIII(a)	**Explain** **Example**	5 4		

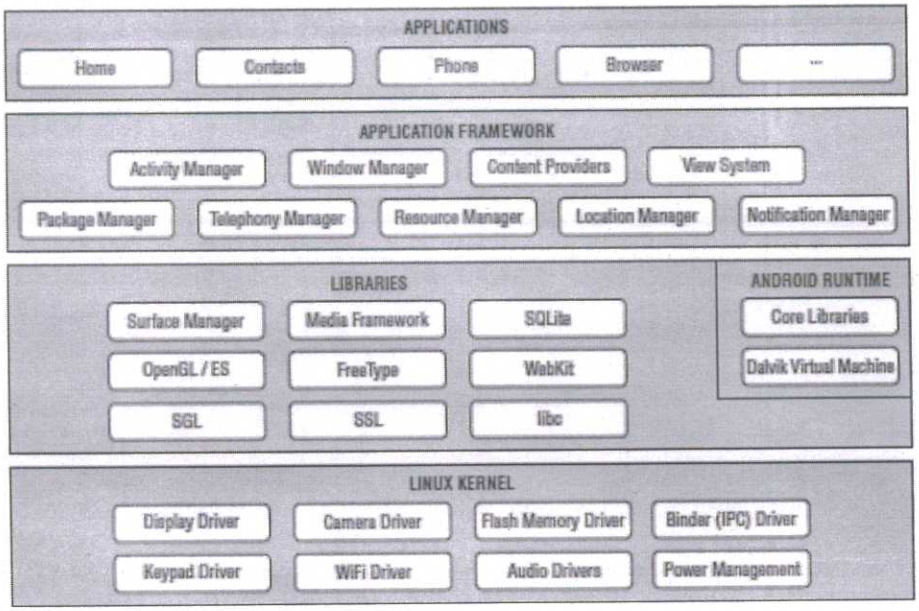
	Database - Package
	
	The main package is android.database.sqlite that contains the classes to manage your own databases
	Database - Creation
	In order to create a database you just need to call this method openOrCreateDatabase with your database name and mode as a parameter. It returns an instance of SQLite database which you have to receive in your own object.Its syntax is given below
	SQLiteDatabase mydatabase = openOrCreateDatabase("your database name",MODE_PRIVATE,null);
	Database - Insertion
	we can create table or insert data into table using execSQL method defined in SQLiteDatabase class. Its syntax is given below
	mydatabase.execSQL("CREATE TABLE Table_name(Columns type);");
	mydatabase.execSQL("INSERT INTO Table_name VALUES(values);");
	
	Database - Helper class
	
	For managing all the operations related to the database , an helper class has been given and is called SQLiteOpenHelper. It automatically manages the creation and update of the database. Its syntax is given below
	public class DBHelper extends SQLiteOpenHelper {
	public DBHelper(){
	super(context,DATABASE_NAME,null,1);
	}
	public void onCreate(SQLiteDatabase db) {}
	public void onUpgrade(SQLiteDatabase database, int oldVersion, int newVersion) {}
	}
			9	

VIII(b)	To save text into a file, you use the FileOutputStream class. The openFileOutput() method opens a named file for writing, with the mode specified. You use the MODE_WORLD_READABLE constant to indicate that the file is readable by all other applications: FileOutputStream fOut = openFileOutput("textfile.txt", MODE_WORLD_READABLE); To convert a character stream into a byte stream, you use an instance of the OutputStreamWriter class, by passing it an instance of the FileOutputStream object: OutputStreamWriter osw = new OutputStreamWriter(fOut);	2 2	6	

	You then use its write() method to write the string to the file. To ensure that all the bytes are written to the file, use the flush() method. Finally, use the close() method to close the file:
	osw.write(str); // str is of type String
	osw.flush();
	osw.close();
		2		

IX	List 3 ways
	attach it to a web page in one of three ways:
	.(1) Inline in the tags themselves
	.(2) Embedded in the head of your HTML
	.(3) In a separate document as an external style sheet
	(1) Styles that are placed inline inside a tag don't need a selector because the selector is defined by the tag it's in. You add a style attribute to the tag, and put the styles in the attribute value (separate multiple styles by a semicolon (;). For example, to color the text of a single paragraph in your HTML red you would write:
	
	(2) Embedded style sheets sit in the <head> tag of your document. You use the <style>
	tag and write your styles as mentioned previously with a selector and styles enclosed
	in curly braces. For example, here is a simple page with paragraphs in red text:
	<!DOCTYPE html>
	<html>
	<head>
	<title>Example of Embedded Styles</title>
	<style>
	p {color: red;
	}
	</style>
	</head>
	<body>
	 The text in this paragraph would be red.
	 And this paragraph as well.
	</body>
	</html>
	
	(3) To create an external style sheet:
	1. Open a new document.
	2. Write your styles as in the earlier embedded style sheet example, but with-
	out the <style> tag surrounding them.
	3. Save that file as a style sheet with a .css extension, such as styles.css
	.
	Here is a style sheet with the red paragraphs:
	
	p {
	color: red;
	}
		3
		
		3
		
		4
		
		5
			15	

	Writing the style sheet document is not enough; you have to also attach it to your web page. To do this, add a <link> tag to the head of your document that points to the style sheet. For example: <link href="styles.css" rel="stylesheet">			
X(a)	These tags are: . <article> —An independent portion of the document or site. . <aside> —Content that is tangential to the main part of the page or site. . <figcaption> —Caption for a figure. . <figure> —A figure or quotation pulled out of the flow of text. . <footer> —The footer of a document or section. . <header> —The header of a document or section. . <hgroup> —A group of headings. . <nav> —A navigation section. . <section> —A generic section that cannot be defined by one of the above types.	9*1	9	
X(b)	Explain any three advantages **Compatible on all the platforms** A high level of uniformity can be maintained when the apps are developed for multiple platforms. PhoneGap abolishes the differences in the app's look and feel when viewed on different platforms. **Ease of Development** PhoneGap works on HTML5, CSS3 and JavaScript, the most common browser based skills which do not require any additional inputs. Hence the business owners can utilize their own tech team and get it developed smartly. **Two fold benefits** It's a win win for all the stakeholders. The developers gain as they have to spend less effort and get develop an app which will work across all mobile platforms. The businesses gain as the app is prepared and ready to hit the market in minimalistic time. **Tapping into the device's hardware** One of the most amazing things about this framework is that it taps into the device's hardware such as the camera, geo location, accelerometer and few others. With this, the apps developed on PhoneGap can easily make use of the properties of the native resources of the device, without compromising on the User Experience.	3*2	6	



Architecture –Android (Fig1)

Table1 Compare Android and Apple ioS

	Android	ioS
Source model	Open source	Closed, with open source components.
OS family	Linux	OS X, UNIX
Customizability	A lot. Can change almost anything.	Limited unless jailbroken
Developer	Google, Open Handset Alliance	Apple Inc.
Available language(s)	100+ Languages	34 Languages
File transfer	Easier than iOS. Using USB port and Android File Transfer desktop app. Photos can be transferred via USB without apps.	More difficult. Media files can be transferred using iTunes desktop app. Photos can be transferred out via USB without apps.
Calls and messaging	Google Hangouts. 3rd party apps like Facebook Messenger, WhatsApp, Google Duo and Skype all work on Android and iOS both.	iMessage, FaceTime (with other Apple devices only).
Internet browsing	Google Chrome (or Android Browser on older versions; other browsers are available)	Mobile Safari (Other browsers are available)

App store , Affordability and interface	Google Play – 1,000,000+ apps. Other app stores like Amazon and Getjar also distribute Android apps.	Apple app store – 1,000,000+ apps
Video chat	Google Duo and other 3rd party apps	FaceTime (Apple devices only) and other 3rd party apps
Voice commands	Google Now, Google Assistant	Siri
Maps	Google Maps	Apple Maps

Diagram 2 Life cycle of activity