

## SCHEME OF VALUATION

### (Scoring Indicators)

| Revision : 2015                         |   | Course   |           |       |
|---|---|--|-----------|-------|
| Code : 6133                             |   |  |           |       |
| Course Title : SMART DEVICE PROGRAMMING |   |  |           |       |
| Qn. No.                                 | Scoring Indicator   | Split up Score                                 | Sub Total | Total |
| <b>I. PART A</b>                        |   |  |           |       |
| 1                                       | Eclipse<br>Android Studio   | 1+1  | 2         | 2     |
| 2.                                      | Activity is a window that contains the user interface of Android applications, which may contain widgets like buttons, labels, text boxes, and so on.   | 2  | 2         | 2     |
| 3                                       | A Bundle object is basically a dictionary object that enables you to set data in key/value pairs.   | 2  | 2         | 2     |
| 4                                       | A view is a widget that has an appearance on screen. Examples of views are buttons, labels, and text boxes.   | 1 + 1 (two example)                            | 2         | 2     |
| 5                                       | It is a framework or a tool to help you create mobile applications, and also works to convert HTML5 applications into native mobile apps.   | 2  | 2         | 2     |
| <b>II. PART B</b>                       |   |  |           |       |
| 1.                                      | GSM (Global System for Mobile communication), GPRS (General Packet Radio Service), EDGE (Enhanced Data rates for GSM Evolution), UMTS (Universal Mobile Telecommunications System), CDMA (Code Division Multiple Access), 1G, 2G, 3G, 4G, 5G, LTE (Long Term Evolution)   | List – 2 marks + explanation any two (4) marks | 6         | 6     |
| 2.                                      | <ul style="list-style-type: none"> <li>- The Android Software Development Kit provides all necessary tools to develop Android applications.</li> <li>- It contains tools to create, compile and package Android application.</li> <li>- It also provides an Android device emulator, so Android applications can be tested without a real Android phone.</li> <li>- We can create Android Virtual Devices (AVD) via the Android SDK, which run in this emulator.</li> <li>- The Android SDK contains the Android Debug Bridge (adb) tool which allows to connect to a virtual or real Android device.</li> <li>- Android SDK contains a debugger, libraries, an emulator, documentation, sample code, and tutorials.</li> </ul> | 6  | 6         | 6     |
| 3.                                      | An <u>Intent</u> is the “glue” that enables different activities from different applications to work together, ensuring that tasks  | 2+2+2  | 6         | 6     |

|    |  |                                       |   |   |
|----|--|---------------------------------------|---|---|
|    | <p>can be performed as though they all belong to one single application. It is a messaging object you can use to request an action from another app component.</p> <p><u>Explicit Intent</u> – specify which application will satisfy the intent, by supplying either the target app’s package name or fully-qualified component class name.</p> <p><u>Implicit Intent</u> – do not name a specific component, instead declare a general action to perform, which allows a component from another app to handle it.</p>  |                                       |   |   |
| 4. | <p>A content provider is the way to share data across packages.</p> <p>A content provider behaves very much like a database — you can query it, edit its content, as well as add or delete its content.</p> <p>Unlike a database, a content provider can use different ways to store its data. The data can be stored in a database, in files, or even over a network.</p> <p>Many useful content providers are,</p> <p><b>Browser</b> — Stores data such as browser bookmarks, browser history, and so on</p> <p><b>CallLog</b> — Stores data such as missed calls, call details.</p> <p><b>Contacts</b> — Stores contact details</p> <p><b>MediaStore</b> — Stores media files such as audio, video and images</p> <p><b>Settings</b> — Stores the device’s settings and preferences</p>   | 2 + 4 (Any 4 x 1)                     | 6 | 6 |
| 5. | <p>To send an SMS message programmatically, use the SmsManager class. No need to directly instantiate this class; instead, call the getDefault() static method to obtain a SmsManager object. Then send the SMS message using the sendTextMessage() method:</p> <pre>private void sendSMS(String phoneNumber, String message)</pre> <pre>{ SmsManager sms = SmsManager.getDefault(); sms.sendTextMessage(phoneNumber, null, message, null, null); }</pre> <p>Five arguments to the sendTextMessage() method:</p> <ul style="list-style-type: none"> <li>▶ destinationAddress — Phone number of the recipient</li> <li>▶ scAddress — Service center address; use null for default.</li> <li>▶ text — Content of the SMS message</li> <li>▶ sentIntent — Pending intent to invoke when the message is sent</li> <li>▶ deliveryIntent — Pending intent to invoke when the message has been delivered .</li> </ul> | Method (2 Marks) + arguments (4 maks) | 6 | 6 |
| 6. | <p><b>onabort</b>—Fires when an action is aborted.</p> <p><b>onbeforeunload, onbeforeunload, and onunload</b>—</p>   | Any 6 x 1 = 6                         | 6 | 6 |

|               |   |  |   |   |
|---------------|---|--|---|---|
|               | <p>Fires just before an element loads or unloads and as an element unloads.</p> <p><b>oncontextmenu</b>—Fires when the context menu is triggered.</p> <p><b>ondrag, ondragend, ondragenter, ondragleave, ondragstart,</b> and</p> <p><b>ondrop</b>—These fire when various drag-and-drop actions occur.</p> <p><b>onerror</b> and <b>onmessage</b>—These fire when errors or messages are triggered.</p> <p><b>onscroll</b>—This fires when the user scrolls the browser scroll bar.</p> <p><b>onresize</b>—Fires when an element is resized.</p>   |  |   |   |
| 7.            | <p>1. Create your web page as you normally would:<br/> <pre>&lt;!DOCTYPE html&gt; &lt;html&gt; &lt;head&gt; &lt;title&gt;Simple JavaScript&lt;/title&gt; &lt;/head&gt; &lt;body&gt; &lt;/body&gt; &lt;/html&gt;</pre></p> <p>2. Add a script tag to the &lt;head&gt; of the page:<br/> <pre>&lt;script&gt; &lt;/script&gt;</pre></p> <p>3. Add the JavaScript function inside the script tags to write “Hello World” in an alert window:<br/> <pre>function hello() { alert(“Hello World”); }</pre></p> <p>4. Add a link to the body of your document to call the script:<br/> <pre>&lt;a href="#" onclick="hello();"&gt;Click Me&lt;/a&gt;</pre></p> <p>5. Include a &lt;noscript&gt; tag below the link with alternative text:<br/> <pre>&lt;noscript&gt; &lt;p&gt;Hello World &lt;p&gt;This text is not written with JavaScript. &lt;/noscript&gt;</pre></p> | Step (3) + example (3)                         | 6 | 6 |
| <b>PART C</b> |   |  |   |   |
| III.<br>a)    | <p><b>In Windows</b></p> <p>Android programming is Java based and it requires JDK &amp; JRE environment. After installing JDK &amp; JRE, configure environment variables.</p> <pre>Set PATH=C:\jdk1.6.0_15\bin;%PATH% set JAVA_HOME=C:\jdk1.6.0_15</pre> <p>The next step will be installation of Android IDE. After downloading android studio software,</p>   | Installation step in Windows / Linux (9 marks) | 9 | 9 |

|           |   |                  |   |   |
|-----------|---|------------------|---|---|
|           | <p>Step1: Open you downloaded android studio file</p> <p>Step2: It will show you welcome to android studio. - Go for "Next"</p> <p>Step3: Now choose to install, android programming. - Go for "Next"</p> <p>Step4: Go for "I Agree", It's for integrating SDK, it will confirm SDK installation.</p> <p>Step6: Now choose the location for Android Studio and SDK. - Go for "Next"</p> <p>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. This will help your android studio to run faster.</p> <p>Step8: Now choose setting for shortcut of android studio on your computer.</p> <p>Step9: After that it will copy all the files to the storage location you have selected before.</p> <p>Step10. On completion, launch the android studio and configure SDK Manager and AVD.</p> |                  |   |   |
| b)        | <p><u>Customizability</u> : - Android - A lot. Can change almost anything,<br/>iOS - Limited unless jailbroken</p> <p><u>Developer</u> : Android - Google, Open Handset Alliance, iOS - Apple Inc.</p> <p><u>Source model</u> : Android -Open source, iOS- Closed, with open source components.</p> <p><u>OS family</u>:- Android –Linux, iOS - OS X, UNIX</p> <p><u>Widgets</u> :- Android –Yes, iOS- No, except in NotificationCenter</p> <p><u>File transfer</u> :- Android -Easier than iOS, iOS - More difficult. Media files can be transferred using iTunes desktop app.</p> <p><u>Internet browsing</u> : Android - Google Chrome (or Android Browser on older versions; other browsers are available), iOS - Mobile Safari (Other browsers are available)</p>  | Any 6 x 1<br>= 6 | 6 | 6 |
| IV.<br>a) | <p>The AVD Manager is a tool you can use to create update, delete, repair, and manage Android virtual devices (AVDs), which define device configurations for the Android Emulator. To launch the AVD Manager:</p> <ul style="list-style-type: none"> <li>• In Android Studio, select Tools &gt; Android &gt; AVD Manager, or click the AVD Manager icon in the toolbar-&gt;Create AVD</li> <li>• Or, use the command line to navigate to your SDK's tools/ directory and execute: <b>\$ android avd</b></li> <li>• In Eclipse ADT, Below are the steps to create an AVD from AVD manager graphical interface <ul style="list-style-type: none"> <li>○ Go to Window -&gt;AVD Manager and select Virtual Devices.</li> </ul> </li> </ul>  | 9                | 9 | 9 |

|          |   |                              |   |   |
|----------|---|------------------------------|---|---|
|          | <ul style="list-style-type: none"> <li>○ Click on New to create a Virtual Device, give it some Name and select Target Android Platform from the drop down list</li> <li>○ Click “Create AVD”</li> </ul> <p>We can create as many AVDs as you would like to use with the Android Emulator. To effectively test our app, we should create an AVD that models each device type for which we have designed our app to support.</p>  |                              |   |   |
| b)       | <ul style="list-style-type: none"> <li>● Gradle-based build support.</li> <li>● Android-specific refactoring and quick fixes</li> <li>● Lint tools to catch performance, usability, version compatibility and other problems.</li> <li>● A rich layout editor that allows users to drag-and-drop UI components.</li> <li>● Support for building Android Wear apps - designed for smartwatches and other wearables</li> <li>● Android Virtual Device (Emulator) to run and debug apps in the Android studio.</li> <li>● Android Studio supports a number of programming languages.</li> </ul>  | Any 6 x 1<br>= 6             | 6 | 6 |
| V.<br>a) | <ul style="list-style-type: none"> <li>▶ onCreate() — Called when the activity is first created</li> <li>▶ onStart() — Called when the activity becomes visible to the user</li> <li>▶ onResume() — Called when the activity starts interacting with the user</li> <li>▶ onPause() — Called when the current activity is being paused and the previous activity is being resumed</li> <li>▶ onStop() — Called when the activity is no longer visible to the user</li> <li>▶ onDestroy() — Called before the activity is destroyed by the system (either manually or by the system to conserve memory)</li> <li>▶ onRestart() — Called when the activity has been stopped and is restarting again</li> </ul> | Fig (5) +<br>Explanation (4) | 9 | 9 |

|        |  |                 |   |   |
|--------|--|-----------------|---|---|
|        |  |                 |   |   |
| b)     | <p>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 <i>intent</i>.</p> <p>It can be used with <b>startActivity</b> to launch an Activity, <b>broadcastIntent</b> to send it to any interested BroadcastReceiver components, and <b>startService(Intent)</b> or <b>bindService (Intent, ServiceConnection, int)</b> to communicate with a background Service.</p> <p><b>Context.startActivity()</b> - The Intent object is passed to this method to launch a new activity or get an existing activity to do something new.</p> <p><u>MainActivityOne.java</u></p> <pre>Intent i = new Intent(getApplicationContext(), ActivityTwo.class); i.putExtra("Value1", "One"); i.putExtra("Value2", "Two"); startActivity(i);</pre> | 4 + example (2) | 6 | 6 |
| VI. a) | <p><b>Activities</b><br/>An activity represents a single screen with a user interface, in-short Activity performs actions on the screen.</p> <p><b>Services</b><br/>A service is a component that runs in the background to perform long-running operations.</p> <p><b>Broadcast Receivers</b><br/>Broadcast Receivers simply respond to broadcast messages from other applications or from the system.</p> <p><b>Content Providers</b></p>  | 4 x 2 = 8       | 8 | 8 |

|           |  |                               |   |   |
|-----------|--|-------------------------------|---|---|
|           | <p>A content provider component supplies data from one application to other on request. Such requests are handled by the methods of the <i>ContentResolver</i> class. The data may be stored in the file system, the database or somewhere else entirely.</p>  |                               |   |   |
| b)        | <p>It is also common to pass data to an activity. We can use the Intent object to pass the data to the target activity.<br/> <u>Example</u><br/> Intent i = new Intent("net.learn2develop.ACTIVITY2");<br/> Bundle extras = new Bundle();<br/> extras.putString("Name", "Your name here");<br/> i.putExtras(extras);<br/> startActivityForResult(i, 1);<br/> In the target activity, you first use the getIntent() method to obtain the intent that started the activity. You then use the getExtras() method to obtain the Bundle object:<br/> Bundle extras = getIntent().getExtras();<br/> if (extras!=null)<br/> {<br/> defaultName = extras.getString("Name");<br/> }</p>   | Explanation (3) + example (4) | 7 | 7 |
| VII<br>a) | <p><b>LinearLayout</b> : - arranges views in a single column or a single row. Child views can be arranged either vertically or horizontally.<br/> <b>AbsoluteLayout</b> : - enables you to specify the exact location of its children.<br/> <b>TableLayout</b> : - groups views into rows and columns. You use the &lt;TableRow&gt; 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<br/> <b>RelativeLayout</b> : - enables you to specify how child views are positioned relative to each other<br/> <b>FrameLayout</b> : - 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.</p> | Any 4 X 2 = 8                 | 8 | 8 |
| b)        | <p>Menus are useful for displaying additional options that are not directly visible on the main UI of an application. There are two main types of menus in Android:<br/> ► <b>Options menu</b> — Displays information related to the current activity. In Android, you activate the options menu by pressing the MENU key.<br/> ► <b>Context menu</b> — Displays information related to a particular view on an activity. In Android, o activate a</p>   | 2 + 2.5 + 2.5with explanation | 7 | 7 |

|            |   |                                 |   |   |
|------------|---|---------------------------------|---|---|
|            | context menu you tap and hold on to it.   |                                 |   |   |
| VIII<br>a) | <p>SQLite is a opensource SQL database that stores data to a text file on a device. It supports all the relational database features.</p> <p><b>Database creation</b> : To create a database just call the method <b>openOrCreateDatabase</b> with database name and mode as parameter. It returns an instance of SQLite database which have to receive in our own object<br/> <pre>SQLiteDatabase mydatabase = openOrCreateDatabase("dbname",MODE_PRIVATE,null) ;</pre> </p> <p><b>Insertion</b> : - mydatabase.execSQL("CREATE TABLE IF NOT EXISTS poly(username varchar, password varchar);");<br/> Mydatabase.execSQL("INSERT INTO poly VALUES ('admin','123');");</p>  | 4+4 with explanation            | 8 | 8 |
| b)         | <p>TextView - view is used to display text to the user.</p> <p>Button — Represents a push-button widget</p> <p>ImageButton — Similar to the Button view, except that it also displays an image</p> <p>EditText — A subclass of the TextView view, except that it allows users to edit its text content</p> <p>CheckBox — A special type of button that has two states: checked or unchecked</p> <p>RadioGroup and RadioButton — The RadioButton has two states: either checked or unchecked. Once a RadioButton is checked, it cannot be unchecked.</p> <p>A RadioGroup is used to group together one or more RadioButton views, thereby allowing only one RadioButton to be checked within the RadioGroup.</p> <p>ToggleButton — Displays checked/unchecked states using a light indicator</p> | List (2) + explain any five (5) | 7 | 7 |
| IX.<br>a)  | <ol style="list-style-type: none"> <li>1. Compatible on all platforms</li> <li>2. Ease of development</li> <li>3. Two fold benefit</li> <li>4. Tapping into device's hardware</li> <li>5. Strong and robust backend</li> <li>6. Open source</li> <li>7. Flexibility</li> </ol>  | Any 4 x 2                       | 8 | 8 |
| b)         | <p>Most of these new tags are called "sectioning" elements and they provide semantics for the layout and sections of an HTML document.</p> <p>. &lt;article&gt;—An independent portion of the document or site.</p> <p>. &lt;aside&gt;—Content that is tangential to the main part of the page or site.</p> <p>. &lt;figcaption&gt;—Caption for a figure.</p> <p>. &lt;figure&gt;—A figure or quotation pulled out of the flow of text.</p>   | Any seven (7)                   | 7 | 7 |

|          |   |                               |   |   |
|----------|---|-------------------------------|---|---|
|          | <ul style="list-style-type: none"> <li>. &lt;footer&gt;—The footer of a document or section.</li> <li>. &lt;header&gt;—The header of a document or section.</li> <li>. &lt;hgroup&gt;—A group of headings.</li> <li>. &lt;nav&gt;—A navigation section.</li> <li>. &lt;section&gt;—A generic section that cannot be defined by one of the above types.</li> </ul>   |                               |   |   |
| X.<br>a) | <p>To create an external style sheet:</p> <ol style="list-style-type: none"> <li>1. Open a new document.</li> <li>2. Write your styles as in the earlier embedded style sheet example, but without the &lt;style&gt; tag surrounding them.</li> <li>3. Save that file as a style sheet with a .css extension, such as styles.css.</li> </ol> <p>Here is a style sheet with some additional styles along with the red paragraphs:</p> <pre>html, body { margin:0; padding:0; border:0; } body { font: 1em/1.25 Arial, Helvetica, sans-serif; } p { color: red; }</pre> <p>Writing the style sheet document is not enough; you have to also attach it to your web page. To do this, add a &lt;link&gt; tag to the head of your document that points to the style sheet. For example:</p> <pre>&lt;link href="styles.css" rel="stylesheet"&gt;</pre> | Explanation (4) + example (4) | 8 | 8 |
| b)       | <ul style="list-style-type: none"> <li>. &lt;details&gt;—Control for adding more information.</li> <li>. &lt;figcaption&gt;—Caption for a figure.</li> <li>. &lt;figure&gt;—A figure or quotation pulled out of the flow of text.</li> <li>. &lt;mark&gt;—Content that has been highlighted or marked.</li> <li>. &lt;meter&gt;—A scalar gauge.</li> <li>. &lt;output&gt;—Results from a script or form.</li> <li>. &lt;progress&gt;—Progress indicator.</li> <li>. &lt;summary&gt;—Summary or legend for a details element.</li> </ul>   | List (2) + explanation (5)    | 7 | 7 |