

**SMART DEVICE PROGRAMMING**

TED (15): 6133

| Qn. No.       | Scoring Indicators  | Split score | Total score |
|---------------|---|-------------|-------------|
| <b>PART A</b> |   |             |             |
| 1             | Android OS, Bada, Blackberry, IOS, Symbian OS, Windows mobile (write any four).   | 2           | 2           |
| 2             | An Android Virtual Device (AVD) is an emulator configuration that allows developers to test the application by simulating the real device capabilities.   | 2           | 2           |
| 3             | An activity is a window that contains the user interface of your application. An application can have zero or more activities. The main purpose of an activity is to interact with user.  | 2           | 2           |
| 4             | Linear, Relative, Absolute, Table, Frame (Any 4)  | 4 x 0.5     | 2           |
| 5             | onload(), onFocus(), onError(), onclick() : Any 4   | 4x 0.5      | 2           |
| <b>PART B</b> |   |             |             |
| 1             | <ul style="list-style-type: none"> <li>• Messaging</li> <li>• Web browser</li> <li>• Voice-based features</li> <li>• Multi-touch</li> <li>• Multitasking</li> <li>• Screen capture</li> <li>• TV recording</li> <li>• Video calling</li> <li>• Multiple language support</li> <li>• Accessibility</li> <li>• Storage</li> <li>• Wi-Fi Direct</li> </ul>   | 6 x 1       | 6           |
| 2             | <p>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</li> <li>• Go to Window -&gt;AVD Manager and select Virtual Devices.</li> <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> | 6 x 1       | 6           |
| 3             | <p><b>Implicit Intent</b></p> <ul style="list-style-type: none"> <li>• Implicit intents specify the action which should be performed and optionally data for the action.</li> <li>• If an implicit intent is sent to the Android system, it searches for all components which are registered for the specific action and the fitting data type.</li> <li>• If only one component is found, Android starts this component directly.</li> </ul>   |             |             |

|   |   |       |   |
|---|---|-------|---|
|   | <ul style="list-style-type: none"> <li>If several components are identified by the Android system, the user will get a selection dialog and can decide which component should be used for the intent. This process is called <b>intent resolution</b>.</li> <li>Intent filters are used for this which is specified in AndroidManifest.xml</li> <li>For example, the following tells the Android system to view a webpage. <ul style="list-style-type: none"> <li>All installed web browsers should be registered to the corresponding intent data via an intent filter. <pre>Intent i=new Intent(Intent.ACTION_VIEW); i.setData(Uri.parse("http://www.google.com")); startActivity(i);</pre> </li> </ul> </li> </ul> <p><b>Explicit Intent</b></p> <ul style="list-style-type: none"> <li><b>Explicit Intent</b> specifies the component.</li> <li>In such case, intent provides the external class to be invoked.</li> <li>Explicit intents explicitly define the component which should be called by the Android system, by using the Java class as identifier.</li> <li>The following shows how to create an explicit intent and send it to the Android system to start an activity. <pre>Intent i = new Intent(MainActivity.this, SecondActivity.class); startActivity(i);</pre> </li> </ul> | 3 x 2 | 6 |
| 4 | <p><b>Steps to write</b></p> <ul style="list-style-type: none"> <li>getExternalStorageDirectory() method is used to return the full path to the external storage.</li> <li><code>File sdCard = Environment.getExternalStorageDirectory();</code></li> <li>Create a new directory</li> <li><code>File directory = new File (sdCard.getAbsolutePath() + "/MyFiles");</code></li> <li>Finally, you save the file into this directory.</li> <li><code>File file = new File(directory, "textfile.txt");</code></li> <li>Continue similar to internal storage writing as follows <pre>OutputStreamWriter osw = new OutputStreamWriter(fOut); osw.write(str); osw.flush(); osw.close();</pre> </li> </ul> <p>In order to write to the external storage, we need to add the WRITE_EXTERNAL_STORAGE permission in your AndroidManifest.xml file:</p> <pre>&lt;uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"&gt; &lt;/uses-permission&gt;</pre> <p><b>Steps to read</b></p> <p>To load the file from the external storage, modify the onClick() method for the Load button:</p> <pre>File sdCard = Environment.getExternalStorageDirectory();</pre>  | 3 x 2 | 6 |

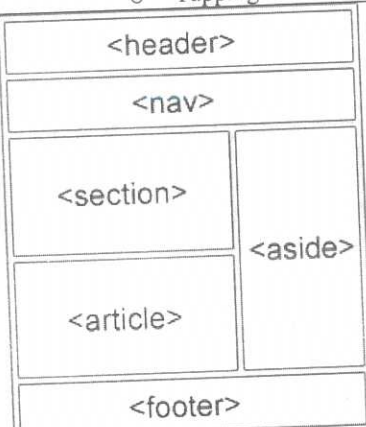
|   |  |       |   |
|---|--|-------|---|
|   | <pre>File directory = new File (sdCard.getAbsolutePath() +"/MyFiles"); File file = new File(directory, "textfile.txt"); FileInputStream fln = new FileInputStream(file); InputStreamReader isr = new InputStreamReader(fln);  Remaining code is similar to reading from internal storage.ie. char[] inputBuffer =newchar[READ_BLOCK_SIZE]; String s = ""; intcharRead; while ((charRead = isr.read(inputBuffer))&gt;0) {String readString = String.valueOf(inputBuffer, 0,charRead); s += readString; inputBuffer = new char[READ_BLOCK_SIZE];}</pre>  |       |   |
| 5 | <p><b>Database creation:</b></p> <p>Create a database with name "MyDb" with version number "1". Also Create a table "user" with field (id, username, email and password). Steps are:</p> <ol style="list-style-type: none"> <li>Create a new class (DataBase) as sub set of "SQLiteOpenHelper" class</li> <li>Set version number(1)</li> <li>Call the context of this class (DataBase) to create database(MyDb)</li> <li>In onCreate method, create table making query, and call execSQL to create the table</li> </ol> <p><b>Insert data into SQLitedatabase:</b></p> <p>For inserting to database steps are:</p> <ol style="list-style-type: none"> <li>Declare database object</li> <li>Declare object of DataBase class and open the database.</li> <li>Load the values from activity into variables as required data type</li> <li>Create object of ContentValues class</li> <li>Use "put method" to map values to fields values.</li> <li>use insert method to insert a row of data</li> </ol> |       |   |
| 6 | <p><b>Inline CSS:</b> An inline style may be used to apply a unique style for a single HTML element. To use inline styles, add the style attribute to the relevant element. The style attribute can contain any CSS property.<br/>Eg: &lt;h1 style="color : red ; align : centre&gt; Inline CSS &lt;/h1&gt;</p> <p><b>Internal CSS:</b> An internal style sheet may be used if one single page has a unique style. Internal styles are defined in the &lt;head&gt; section of the HTML page inside the &lt;style&gt; tag.</p> <p><b>External CSS:</b> An external style sheet is ideal when the style is applied to many pages. With an external style sheet, you can change the look of an entire Web site by changing one file. Each page must link to the style sheet using the &lt;link&gt; tag. To create an external style sheet:</p> <p>Open a new text editor</p> <p>Define all the Style rules within this text file.</p>   | 3 x 2 | 6 |

|               |   |       |   |
|---------------|---|-------|---|
|               | Save that file as a style sheet with a .css extension, such as example.css .<br>Include this file in the head section of HTML document using <link> element.  |       |   |
| 7             | <p><b>Working with Media Files</b></p> <p>In PhoneGap, the Media Capture API is used for capturing photos, audio and video data as well. The Media Capture API allows you to start and stop recording, play, pause, and stop media files, and even display an audio file's duration. PhoneGap will play locally saved files, as well as those on a remote web server.</p> <p>The PhoneGap Media object has three parameters –src, mediaSuccess, mediaError .</p> <p>The methods allow you to play, record, and pause, among other things. Methods are: getCurrentPosition(), getDuration(), play(), pause(), release(), startRecord(), stop()</p>   |       |   |
| <b>PART C</b> |   |       |   |
| III (a)       | <p><b>AndroidManifest.xml</b></p> <ol style="list-style-type: none"> <li><b>&lt;manifest&gt;</b> - manifest is the root element of the AndroidManifest.xml file. It has package attribute that describes the package name of the activity class and also can specify the version code and version name of your application. xmlns specifies the xml namespace of a resource.</li> <li><b>&lt;uses sdk&gt;</b> - Defines minimum sdk version and target sdk version</li> <li><b>&lt;application&gt;</b> - application is the sub element of the manifest. It includes the namespace declaration. This element contains several sub elements that declares the application component such as activity etc. The commonly used attributes are of this element are <b>icon, label, theme</b> etc.</li> <li><b>&lt;activity&gt;</b> - activity is the subelement of application and represents an activity that must be defined in the AndroidManifest.xml file. It has many attributes such as label, name, theme, launchMode etc.</li> <li><b>&lt;intent-filter&gt;</b> - intent-filter is the sub-element of activity that describes the type of intent to which activity, service or broadcast receiver can respond to.</li> <li><b>&lt;action&gt;</b> - It adds an action for the intent-filter. The intent-filter must have at least one action element.</li> <li><b>&lt;category&gt;</b> - It adds a category name to an intent-filter.</li> </ol> | 7 x 1 | 7 |
| (b)           | <ul style="list-style-type: none"> <li>• GSM</li> <li>• 1G , 2G , 3G</li> <li>• HSDPA/HSPA</li> <li>• UMTS</li> <li>• EV-DO</li> <li>• LTE</li> <li>• CDMA</li> </ul> <p style="text-align: right;">(Any four)</p>  | 4 x 2 | 8 |
| IV (a)        | <b>Any 8 points</b>   | 8 x 1 | 8 |
| (b)           | <p>Before proceeding, make sure that we have JDK is installed.</p> <p>1. Install the Java Development Kit (JDK)</p> <p>Step1 : \$ sudo add-apt-repository ppa:webupd8team/java<br/>Step2: \$ sudo apt-get update<br/>Step 3: \$ sudo apt-get install oracle-java8-installer oracle-java8-set-default</p>  |       |   |

|        |  |       |   |
|--------|--|-------|---|
|        | <p>Verify if java installation was successful or not by checking the version number<br/> Step4: : \$ java -version<br/> 2. Download Eclipse IDE from its website.<br/> Check out your OS Type, 32-bit or 64-bit, and select Linux 32-bit or 64-bit of Eclipse Installer.<br/> 3. Run the installer wizard.<br/> Decompress the downloaded archive -<br/> <b>Star xf eclipse-inst-linux64.tar.gz</b><br/> Run eclipse-inst -<br/> \$./eclipse-inst<br/> When the wizard launches, select install item and folder, and finally click INSTALL button.<br/> 4. Download the ADT plugin for eclipse<br/> 5. Configuring the ADT plugin<br/> 6. Create an Android Virtual Device (AVD)<br/> 7. Create and run the simple android example</p> | 4+3   | 7 |
| V (a)  | <ol style="list-style-type: none"> <li>1. Activities</li> <li>2. Services</li> <li>3. Broadcast Receivers</li> <li>4. Content Providers</li> </ol>   | 4 x 2 | 8 |
| (b)    |  | 4     | 7 |
|        | Explanation  | 3     |   |
| VI (a) | <p><i>Intent</i> supports three ways to pass data: <b>1. Direct:</b> put our data into intents directly . <b>2. Bundle:</b> create a bundle and set the data here. <b>3. Parcelable:</b> It is a way of “serializing” our object.</p> <p><b>Direct:</b> Intent uses a method called <i>putExtra(String name, .....</i>) that allows us to save inside the Intent our information.</p> <pre>intent i = new Intent(MainActivity.this, SecondActivity.class); i.putExtra("name", RAM); startActivity(i);</pre> <p>In the destination activity, data is retrieved by:</p> <pre>intent i = getIntent();</pre>   |       |   |

|         |   |       |    |
|---------|---|-------|----|
|         | <p>String n = i.getStringExtra("name");</p> <p><b>Bundle:</b> Android has a class called <b>Bundle</b> where we can store our data. It supports several data types like strings, chars, boolean, integer and so on. We store our information directly into the <b>bundle</b> and then save the bundle into the Intent.</p> <pre>Intent i = new Intent(MainActivity.this, SecondActivity.class); Bundle b = new Bundle(); b.putString("name", RAM); b.putString("roll", CT1); i.putExtras( b); startActivity(i);</pre> <p>Notice that , we don't store our data into the Intent but into the Bundle, while in the destination activity:</p> <pre>Intent in = getIntent(); Bundle b = i.getExtras(in); String name = b.getString("name"); String surname = b.getString("roll");</pre> <p><b>Parcelable:</b> This is the most elegant way to pass data between activities. <i>Parcelable</i> in android is an interface and every object that wants to be passed across different activities using intent has to implement this interface. This interface has two method that we have to implements:</p> <p><i>describeContent()</i> that returns an <i>int</i></p> <p><i>writeToParcel(Parcel dest, int flags)</i> that returns a void.</p> <p>More over a class that implements this interface must have a static field called <b>CREATOR</b> that is used by the OS to recreate the object.</p> | 3 x 5 | 15 |
| VII (a) | <ul style="list-style-type: none"> <li>• TextView</li> <li>• EditText</li> <li>• Button</li> <li>• ImageButton</li> <li>• CheckBox</li> <li>• ToggleButton</li> <li>• RadioButton</li> <li>• RadioGroup</li> </ul>  | 8 x 1 | 8  |
| (b)     | <ul style="list-style-type: none"> <li>• In the AndroidManifest.xml file, we can add following two permissions – SEND_SMS and RECEIVE_SMS whichever is needed.:</li> </ul> <pre>&lt;uses-permission android:name="android.permission.SEND_SMS"&gt; &lt;/uses-permission&gt; &lt;uses-permission android:name="android.permission.RECEIVE_SMS"&gt; &lt;/uses-permission&gt;</pre>  |       |    |

|          |  |       |   |
|----------|--|-------|---|
|          | <ul style="list-style-type: none"> <li>In the main.xml file located in the res/layout folder, add code so that the user can enter a phone number as well as a message to send using edittexts and</li> <li>Message is send using the sendSMS() function button.</li> <li>sendSMS is defined as below:<br/> <pre>private void sendSMS(String phoneNumber, String message) { SmsManager sms = SmsManager.getDefault();   sms.sendTextMessage(phoneNumber, null, message, null, null);}</pre> </li> <li>Five arguments to the sendTextMessage() method:<br/> destinationAddress : Phone number of the recipient<br/> srcAddress: Service center address; use null for default SMSC<br/> text: Content of the SMS message<br/> sentIntent: Pending intent to invoke when the message is<br/> senddeliveryIntent: Pending intent to invoke when the message has been delivered</li> </ul>   | 7     | 7 |
| VIII (a) | <ol style="list-style-type: none"> <li>Create an account</li> <li>Familiarise yourself with DeveloperConsole</li> <li>Fill in the necessary account details</li> <li>Link your merchant account</li> <li>Upload your app</li> <li>Alpha and beta testing of app</li> <li>Provide details for store listing</li> <li>Add pricing and distribution details</li> <li>Publishing the application</li> </ol>  | 9 x 1 | 9 |
| (b)      | <p><b>Advantages of content providers</b></p> <ul style="list-style-type: none"> <li>Content provider helps to share data across applications.</li> <li>You can configure a content provider to allow other applications to securely access and modify your app data.</li> <li>They encapsulate the data, and provide mechanisms for defining data security</li> <li>Content providers offer granular control over the permissions for accessing data. : You can choose to restrict access to a content provider from solely within your application, grant blanket permission to access data from other applications, or configure different permissions for reading and writing data.</li> <li>You can use a content provider to abstract away the details for accessing different data sources in your application. For example, your application might store structured records in a SQLite database, as well as video and audio files. You can use a content provider to access all of this data, if you implement this development pattern in your application.</li> </ul> <p><b>Disadvantages of content providers</b></p> <ul style="list-style-type: none"> <li>Content providers can only use SQLite as a storage mechanism.</li> <li>If using object-oriented databases, a proxy could be used between the content provider and the database interface.; Or a temporary SQLite database could be a solution</li> <li>Resources are crucial on a mobile environment and you might be better off using SQLite.</li> <li>More complicated queries, with joins and distinct, are not possible with content provider.</li> </ul> | 4     | 7 |
|          |  | 3     |   |

|        |  |         |   |
|--------|--|---------|---|
| IX (a) | <ul style="list-style-type: none"> <li>• JavaScript can be placed in three places of the HTML document.</li> <li>➤ Between the &lt;head&gt; tag of html</li> <li>➤ Between the &lt;body&gt; tag of html</li> <li>➤ In .js file (external javaScript)</li> </ul>  | 3 x 3   | 9 |
| (b)    | <ul style="list-style-type: none"> <li>○ Strong and Robust Backed:</li> <li>○ Open Source:</li> <li>○ Flexibility:</li> <li>○ Compatible on all the platforms</li> <li>○ Ease of Development:</li> <li>○ Two fold benefits:</li> <li>○ Tapping into the device's hardware:</li> </ul>  | 6 x 1   | 6 |
| X (a)  |  <p>The diagram illustrates the structure of an HTML5 document. It consists of several nested boxes representing HTML5 elements: <ul style="list-style-type: none"> <li>A top box labeled <code>&lt;header&gt;</code>.</li> <li>A box below it labeled <code>&lt;nav&gt;</code>.</li> <li>A box below that, which is split into two columns. The left column contains <code>&lt;section&gt;</code> and <code>&lt;article&gt;</code> stacked vertically. The right column contains <code>&lt;aside&gt;</code>.</li> <li>A bottom box labeled <code>&lt;footer&gt;</code>.</li> </ul> </p> | 6 x 1.5 | 9 |
| (b)    | <ul style="list-style-type: none"> <li>• &lt;audio&gt;</li> <li>• &lt;video&gt;</li> <li>• &lt;source&gt;</li> <li>• &lt;track&gt;</li> <li>• &lt;canvas&gt;</li> <li>• &lt;embed&gt;</li> </ul>   | 6 x 1   | 6 |

CODE:6133

COURSE: SMART DEVICE PROGRAMMING

### Questionwise Analysis

| Qn #            | Specific Outcome | Module | Content Details                      | Score | Time in Mts |
|-----------------|------------------|--------|--------------------------------------|-------|-------------|
| <b>Part A</b>   |                  |        |                                      |       |             |
| 1               | 1.1.1            | I      | Mobile operating systems             | 2     | 2           |
| 2               | 2.1.1            | II     | How AVD operates                     | 2     | 2           |
| 3               | 2.1.2            | II     | Activity                             | 2     | 2           |
| 4               | 3.1.1            | III    | UI components                        | 2     | 2           |
| 5               | 4.1.3            | IV     | HTML 5 tags                          | 2     | 2           |
| <b>Part B</b>   |                  |        |                                      |       |             |
| 1               | 1.1.6            | I      | Features of Android                  | 6     | 10          |
| 2               | 2.1.1            | II     | AVD creation                         | 6     | 10          |
| 3               | 2.1.3            | II     | Intent                               | 6     | 10          |
| 4               | 3.2.1            | III    | Storage technologies                 | 6     | 10          |
| 5               | 3.2.2            | III    | SQLite database                      | 6     | 10          |
| 6               | 4.1.1            | IV     | CSS3                                 | 6     | 10          |
| 7               | 4.1.5            | IV     | Build applications using media files | 6     | 10          |
| <b>Part C</b>   |                  |        |                                      |       |             |
| <b>UNIT I</b>   |                  |        |                                      |       |             |
| III a           | 1.1.5            | I      | Manifest file                        | 7     | 15          |
| III b           | 1.1.1            | I      | Mobile technologies                  | 8     | 15          |
| <b>OR</b>       |                  |        |                                      |       |             |
| IV a            | 1.1.2            | I      | Compare IOS and Android              | 8     | 15          |
| IVb             | 1.1.3            | I      | Installation of Eclipse              | 7     | 15          |
| <b>UNIT II</b>  |                  |        |                                      |       |             |
| V a             | 2.1.5            | II     | Application components               | 8     | 15          |
| VI a            | 2.1.2            | II     | Lifecycle of activity                | 7     | 15          |
| <b>OR</b>       |                  |        |                                      |       |             |
| VI b            | 2.1.4            | II     | Data passing between intents         | 15    | 30          |
| <b>UNIT III</b> |                  |        |                                      |       |             |
| VII a           | 3.1.2            | III    | Popular UI components                | 8     | 15          |
| VII B           | 3.3.1            | III    | SMS Service                          | 7     | 15          |
| <b>OR</b>       |                  |        |                                      |       |             |
| VIII a          | 3.3.2            | III    | Publishing application               | 9     | 20          |
| VIII b          | 3.2.3            | III    | Content provider                     | 6     | 10          |
| <b>UNIT IV</b>  |                  |        |                                      |       |             |
| IX a            | 4.1.2            | IV     | Javascript in webpages               | 8     | 15          |
| IXb             | 4.1.4            | IV     | Phonegap technology                  | 7     | 15          |
| <b>OR</b>       |                  |        |                                      |       |             |
| X a             | 4.1.3            | IV     | Multimedia tags                      | 8     | 15          |
| X b             | 4.1.3            | IV     | Event attributes                     | 7     | 15          |

CODE:6133

COURSE: SMART DEVICE PROGRAMMING

**BLUEPRINT**

| Sl. No | Module | Type of questions |       |                  |       |                  |       |                  |       |
|--------|--------|-------------------|-------|------------------|-------|------------------|-------|------------------|-------|
|        |        | Part A            |       | Part B           |       | Part C           |       | Total            |       |
|        |        | No. of Questions  | Score | No. of Questions | Score | No. of Questions | Score | No. of Questions | Score |
| 1      | I      | 1                 | 2     | 1                | 6     | 4                | 30    | 6                | 38    |
| 2      | II     | 2                 | 4     | 2                | 12    | 3                | 30    | 7                | 46    |
| 3      | III    | 1                 | 2     | 2                | 12    | 4                | 30    | 7                | 44    |
| 4      | IV     | 1                 | 2     | 2                | 12    | 4                | 30    | 7                | 44    |
| Total  |        | 5                 | 10    | 7                | 42    | 15               | 120   | 27               | 172   |

7