CMR
INSTITUTE OF
TECHNOLOGY

USN | | | | | | | | | | |

Internal Assessment Test 3 Answer Key– January. 2022									
Sub:	Mobile Applications					Sub Code:	18MCA52	Branch:	MCA
Date:	24/1/2022	Duration:	90 min's	Max Marks:	50	Sem	V		

Q1) What are the different methods for getting location data? Explain

Nowadays, mobile devices are commonly equipped with **GPS receivers**

- because of the many satellites orbiting the earth; you can use a GPS receiver to find your location easily.
- However, GPS requires a clear sky to work and hence does not always work indoors or where satellites can't penetrate (such as a tunnel through a mountain).

Another effective way to locate your position is through ***cell tower triangulation***.

- When a mobile phone is switched on, it is constantly in contact with base stations surrounding it.
- By knowing the identity of cell towers, it is possible to translate this information into a physical location through the use of various databases containing the cell towers' identities and their exact geographical locations.
- The advantage of cell tower triangulation is that it works indoors, without the need to obtain information from satellites. It is not as precise as GPS because its accuracy depends on overlapping signal coverage, which varies quite a bit.
- Cell tower triangulation works best in densely populated areas where the cell towers are closely located.

A third method of locating your position is to rely on **Wi-Fi triangulation.**

- Rather than connect to cell towers, the device connects to a Wi-Fi network and checks the service provider against databases to determine the location serviced by the provider

On the Android, the SDK provides the LocationManager class to help your device determine the user's physical location.

```
lm.requestLocationUpdates( LocationManager.GPS_PROVIDER, 0, 0, locationListener);
```

This method takes four parameters:

1. Provider -The name of the provider with which you register. In this case, you are using GPS to obtain your geographical location data.
2. minTime - The minimum time interval for notifications, in milliseconds.
3. minDistance - The minimum distance interval for notifications, in meters.
4. Listener - An object whose onLocationChanged() method will be called for each location update

Q2) Discuss APK file deployment in detail

Once you have signed your APK files, you need a way to get them onto your users' devices. Three methods are here:

- Deploying manually using the adb.exe tool
- Hosting the application on a web server
- Publishing through the Android Market

Besides the above methods, you can install your applications on users' devices through emails, SD card, etc. As long as you can transfer the APK file onto the user's device, you can install the application.

Using the adb.exe Tool:
Once your Android application is signed, you can deploy it to emulators and devices using the adb.exe (Android Debug Bridge) tool (located in the platform-tools folder of the Android SDK). Using the command prompt in Windows, navigate to the "<Android_SDK>\platform-tools" folder. To install the application to an emulator/device (assuming the emulator is currently up and running or a device is currently connected), issue the following command:

adb install "C:\Users\Wei-Meng Lee\Desktop\LBS.apk"
(Note that, here, LBS is name of the project)

Besides using the adb.exe tool to install applications, you can also use it to remove an installed application. To do so, you can use the shell option to remove an application from its installed folder:

adb shell rm /data/app/net.learn2develop.LBS.apk

Another way to deploy an application is to use the DDMS tool in Eclipse. With an emulator (or device) selected, use the File Explorer in DDMS to go to the /data/app folder and use the "Push a file onto the device" button to copy the APK file onto the device.

Using a Web Server:
If you wish to host your application on your own, you can use a web server to do that. This is ideal if you have your own web hosting services and want to provide the application free of charge to your users or you can restrict access to certain groups of people. Following are the steps involved:

- Copy the signed LBS.apk file to c:\inetpub\wwwroot\. In addition, create a new HTML file named Install.html with the following content:

```
<html>
    <title>Where Am I application</title>
    <body>
        Download the Where Am I application <a href="LBS.apk">here</a>
    </body>
</html>
```

- On your web server, you may need to register a new MIME type for the APK file. The MIME type for the .apk extension is application/vnd.android.packagearchive.
- From the Application settings menu, check the "Unknown sources" item. You will be prompted with a warning message. Click OK. Checking this item will allow the Emulator/device to install applications from other non-Market sources (such as from a web server).
- To install the LBS.apk application from the IIS web server running on your computer, launch the Browser application on the Android Emulator/device and navigate to the URL pointing to the APK file. To refer to the computer running the emulator, you should use the special IP address of 10.0.2.2.
- Alternatively, you can also use the IP address of the host computer. Clicking the "here" link will download the APK file onto your device. Drag the notification bar down to reveal the download status. To install the downloaded application, simply tap on it and it will show the permission(s) required by this application.
- Click the Install button to proceed with the installation. When the application is installed, you can launch it by clicking the Open button.

Besides using a web server, you can also

- email your application to users as a attachment; when the users receive the e-mail they can download the attachment and install the application directly onto their device.

Publishing on Android Market:
It is always better to host your application on Android market (Google Playstore). Steps involved in doing so, are explained hereunder:

- **Creating a Developer Profile:**
  - Create a developer profile at http://market.android.com/publish/Home using a Google account.
  - Pay one-time registration fees.
  - Agree Android Market Developer Distribution Agreement

- **Submitting Your Apps:**
  If you intend to charge for your application, click the Setup Merchant Account link located at the bottom of the screen. Here you enter additional information such as bank account and tax ID. You will be asked to supply some details for your application. Following are the compulsory details to be provided:
  - The application in APK format
  - At least two screenshots. You can use the DDMS perspective in Eclipse to capture screenshots of your application running on the Emulator or real device. A high-resolution application icon. This size of this image must be 512×512 pixels.
  - Provide the title of your application, its description and recent update details.
  - Indicate whether your application employs copy protection, and specify a content rating.

When all these setup is done, click Publish to publish your application on the Android Market.

Q3) Explain the concept of sending SMS and write the procedure for sending an SMS through Android application with a code segment.

SMS messaging is one of the main *killer applications* on a mobile phone today — for some users as necessary as the phone itself. Any mobile phone you buy today should have at least SMS messaging capabilities, and nearly all users of any age know how to send and receive such messages. Android comes with a built-in SMS application that enables you to send and receive SMS messages. However, in some cases you might want to integrate SMS capabilities into your own android application. For example, you might want to write an application that automatically sends a SMS message at regular time intervals. For example, this would be useful if you wanted to track the location of your kids — simply give them an Android device that sends out an SMS message containing its geographical location every 30 minutes.

Sending SMS Messages Programmatically
To create an application that can send SMS, following are the steps to be followed:

- Create a new android application.
- Add the following statements in to the main.xml file:

```
<Button
android:id=”@+id/btnSendSMS”
android:layout_width=”fill_parent”
android:layout_height=”wrap_content”
android:text=”Send SMS” />
```

- In the AndroidManifest.xml file, add the following statements:

```
<uses-sdk android:minSdkVersion=”8” />
<uses-permission android:name=”android.permission.SEND_SMS”>
</uses-permission>
```

- Add the following statements to the MainActivity.java file:

```
import android.app.PendingIntent;
import android.content.Intent;
import android.telephony.SmsManager;
import android.view.View;
```

```
import android.widget.Button;
public class MainActivity extends Activity
{
Button btnSendSMS;
/** Called when the activity is first created.
@Override
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
btnSendSMS = (Button) findViewById(R.id.btnSendSMS);
btnSendSMS.setOnClickListener(new View.OnClickListener()
{
public void onClick(View v)
{
sendSMS("5556", "Hello my friends!");
}
});
}
private void sendSMS(String phoneNumber, String message)
{
SmsManager sms = SmsManager.getDefault();
sms.sendTextMessage(phoneNumber, null, message, null, null);
}
}
```

Following are the five arguments to the sendTextMessage() method:

- destinationAddress — Phone number of the recipient
- scAddress — Service center address; use null for default SMSC
- text — Content of the SMS message
- sentIntent — Pending intent to invoke when the message is sent
- deliveryIntent — Pending intent to invoke when the message has been Delivered

Q4) Explain the concept of sending email and write the procedure for sending an email through Android application with a code segment

One can set email through Android program. Following are the steps involved:

 Create a new android application.

 Add the following statements in to the main.xml file:

```
<Button android:id="@+id/btnSendEmail" android:layout_width="fill_parent"
android:layout_height="wrap_content" android:text="Send Email" />
```

 Add the following statements in bold to the MainActivity.java file:

```
import android.content.Intent; import android.net.Uri;
import android.view.View; import android.widget.Button;
public class MainActivity extends Activity
{
Button btnSendEmail;
/** Called when the activity is first created. */ @Override
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState); setContentView(R.layout.main);
btnSendEmail = (Button) findViewById(R.id.btnSendEmail);
```

```
btnSendEmail.setOnClickListener(new View.OnClickListener()
{
public void onClick(View v)
{
}
});
String[] to = ,“weimenglee@learn2develop.net”,
“weimenglee@gmail.com”-; String*+ cc = ,“course@learn2develop.net”-; sendEmail(to,cc,“Hello”, “Hello
my friends!”);
}
//---sends an SMS message to another device---
private void sendEmail(String[] emailAddresses, String[] carbonCopies, String subject, String message)
{
Intent emailIntent = new Intent(Intent.ACTION_SEND); emailIntent.setData(Uri.parse(“mailto:”));
String[] to = emailAddresses; String[] cc = carbonCopies;
emailIntent.putExtra(Intent.EXTRA_EMAIL, to); emailIntent.putExtra(Intent.EXTRA_CC, cc);
emailIntent.putExtra(Intent.EXTRA_SUBJECT, subject); emailIntent.putExtra(Intent.EXTRA_TEXT,
message); emailIntent.setType(“message/rfc822”); startActivity(Intent.createChooser(emailIntent,“Email”));
}
}
```

 Test the application by running it. When you click on *Send Mail* button you can see Email application launched on device as shown below –

Q5.Write in details how to display Google maps in your own android applications

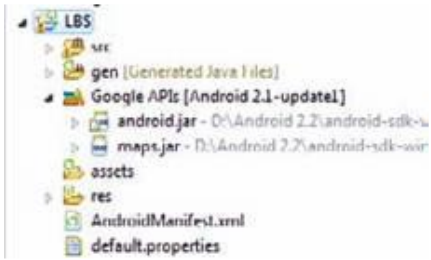
Google Maps is one of the many applications bundled with the Android platform. In addition to simply using the Maps application, you can also embed it into your own applications and make it do some very cool things. This section describes how to use Google Maps in your Android applications and programmatically perform the following:

- Change the views of Google Maps.
- Obtain the latitude and longitude of locations in Google Maps.
- Perform geocoding and reverse geocoding (translating an address to latitude and longitude and vice versa).
- Add markers to Google Maps.

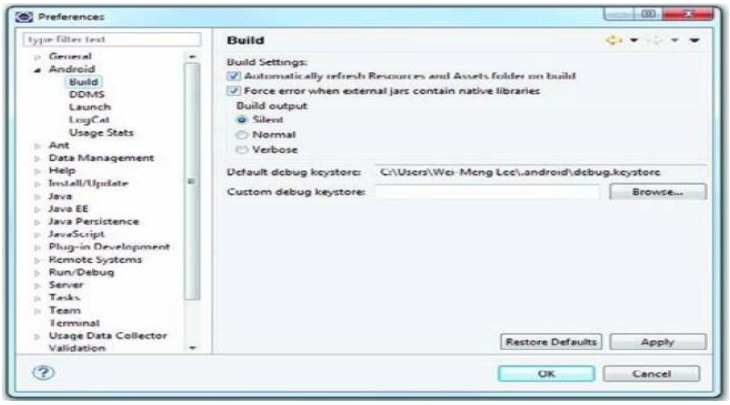
We will discuss how to build a project using maps.

Creating the Project: Create a new android project. In order to use Google Maps in your Android application, you need to ensure that you check the Google APIs as your build target. Google Maps is not

part of the standard Android SDK, so you need to find it in the Google APIs add-on. If LBS is the name of your project, then you can see the additional JAR file (maps.jar) located under the Google APIs folder as below–



Obtaining the Maps API Key: Beginning with the Android SDK release v1.0, you need to apply for a free Google Maps API key before you can integrate Google Maps into your Android application. When you apply for the key, you must also agree to Google's terms of use, so be sure to read them carefully.



First, if you are testing the application on the Android Emulator or an Android device directly connected to your development machine, locate the SDK debug certificate located in the default folder (C:\Users\<username>\.android for Windows 7 users).
You can verify the existence of the debug certificate by going to Eclipse and selecting
Window ⇨Preferences.
Expand the Android item and select Build (as shown in figure above). On the right side of the window, you will be able to see the debug certificate's location.
The filename of the debug keystore is debug.keystore. This is the certificate that Eclipse uses to sign your application so that it may be run on the Android Emulator or devices.

Q6) Write a note on Asynchronous call in networking?

Connections discussed in the previous sections were all *synchronous* – that is, the connection to a server will not return until the data is received. In real life, this presents some problems due to network connections being inherently slow. When you connect to a server to download some data, the user interface of your application remains frozen until a response is obtained. In most cases, this is not acceptable. Hence, you need to ensure that the connection to the server is made in an asynchronous fashion.
The easiest way to connect to the server asynchronously is to use the AsyncTask class available in the Android SDK. Using AsyncTask enables you to perform background tasks in a separate thread and then return the result in a UI thread. Using this class enables you to perform background operations without needing to handle complex threading issues.
Using the previous example of downloading an image from the server and then displaying the image in an ImageView, you could wrap the code in an instance of the AsyncTask class, as shown below:

```
public class MainActivity extends Activity
{
ImageView img;
private class BackgroundTask extends AsyncTask
<String, Void, Bitmap>
```

```
{
protected Bitmap doInBackground(String... url)
{
Bitmap bitmap = DownloadImage(url[0]); return bitmap;
}
protected void onPostExecute(Bitmap bitmap) { ImageView img = (ImageView) findViewById(R.id.img);
img.setImageBitmap(bitmap);
}
}
private InputStream OpenHttpConnection(String urlString) throws IOException
{
...
}
```

Basically, you defined a class that extends the AsyncTask class. In this case, there are two methods within the BackgroundTask class — doInBackground() and onPostExecute(). You put all the code that needs to be run asynchronously in the doInBackground() method. When the task is completed, the result is passed back via the onPostExecute() method. The onPostExecute() method is executed on the UI thread, hence it is thread safe to update the ImageView with the bitmap downloaded from the server.

To perform the asynchronous tasks, simply create an instance of the BackgroundTask class and call its execute() method:

```
@Override
public void onCreate(Bundle savedInstanceState)
{ super.onCreate(savedInstanceState); setContentView(R.layout.main);
new BackgroundTask().execute(
“http://www.streetcar.org/mim/cable/images/cable-01.jpg”);
}
```

Q7) Briefly discuss the concept of binding activities to the services with a suitable code snippet?

Real-world services are usually more sophisticated, requiring the passing of data so that they can do the job correctly for you. Using the service demonstrated earlier that downloads a set of files, suppose you now want to let the calling activity determine what fi les to download, instead of hardcoding them in the service. Here is what you need to do.

First, in the calling activity, you create an Intent object, specifying the service name:

```
Button btnStart = (Button) findViewById(R.id.btnStartService);
 btnStart.setOnClickListener(new View.OnClickListener()

{
public void onClick(View v)
{
Intent intent = new Intent(getBaseContext(), MyService.class);
}
})
```

You then create an array of URL objects and assign it to the Intent object through its putExtra() method. Finally, you start the service using the Intent object:

```
Button btnStart = (Button) findViewById(R.id.btnStartService); btnStart.setOnClickListener(new
View.OnClickListener()
{
public void onClick(View v)
{
Intent intent = new Intent(getBaseContext(), MyService.class);
try
{
URL[] urls = new URL[]
```

{
new URL("http://www.amazon.com/somefiles.pdf"), new
URL("http://www.wrox.com/somefiles.pdf"), new URL("http://www.google.com/somefiles.pdf"),
new URL("http://www.learn2develop.net/somefiles.pdf")
};
intent.putExtra("URLs", urls);
} catch (MalformedURLException e)
{
e.printStackTrace();
}
startService(intent);
}
});
Note that the URL array is assigned to the Intent object as an Object array. On the service's end, you need to extract the data passed in through the Intent object in the onStartCommand() method:
@Override
public int onStartCommand(Intent intent, int flags, int startId)
{
// We want this service to continue running until it is explicitly
// stopped, so return sticky.
Toast.makeText(this, "Service Started", Toast.LENGTH_LONG).show(); Object*+ objUrls = (Object*+)
intent.getExtras().get("URLs"); URL*+ urls = new URL[objUrls.length];
for (int i=0; i<objUrls.length-1; i++)
{
urls[i] = (URL) objUrls[i];
}
new DoBackgroundTask().execute(urls); return START_STICKY;
}
The preceding first extracts the data using the getExtras() method to return a Bundle object. It then uses the get() method to extract out the URL array as an Object array. Because in Java you cannot directly cast an array from one type to another, you have to create a loop and cast each member of the array individually. Finally, you execute the background task by passing the URL array into the execute() method.
This is one way in which your activity can pass values to the service. As you can see, if you have relatively complex data to pass to the service, you have to do some additional work to ensure that the data is passed correctly. A better way to pass data is to bind the activity directly to the service so that the activity can call any public members and methods on the service directly.

Q8) Briefly discuss the concept of accessing web services in networking

Very often, you need to download XML files and parse the contents (a good example of this is consuming Web services). Therefore, in this section you learn how to connect to a Web service using the HTTP GET method. Once the Web service returns a result in XML, you will extract the relevant parts and display its content using the Toast class.
For this example, the web method you will be using is from http://services.aonaware.com/ DictService/DictService.asmx?op=Define. This web method is from a Dictionary Web service that returns the definitions of a given word.
The web method takes a request in the following format:
GET /DictService/DictService.asmx/Define?**word**=string HTTP/1.1
Host: services.aonaware.com
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
It returns a response in the following format:
<?xml version="1.0" encoding="utf-8"?>
<WordDefinition xmlns="http://services.aonaware.com/webservices/">

<Word>string</Word>
<Definitions>
<Definition>
<Word>string</Word>
<Dictionary>
<Id>string</Id>
<Name>string</Name>
</Dictionary>
<WordDefinition>string</WordDefinition>
</Definition>
<Definition>
<Word>string</Word>
<Dictionary>
<Id>string</Id>
<Name>string</Name>
</Dictionary>
<WordDefinition>string</WordDefinition>
</Definition>
</Definitions>
</WordDefinition>
Hence, to obtain the definition of a word, you need to establish an HTTP connection to the web method and then parse the XML result that is returned.

```
public class MainActivity extends Activity {
ImageView img;
private InputStream OpenHttpConnection(String urlString)
throws IOException
{
//...
}
private Bitmap DownloadImage(String URL)
{
//...
}
private String DownloadText(String URL)
{
//...
}
private void WordDefinition(String word) {
InputStream in = null;
try {
in = OpenHttpConnection(
"http://services.aonaware.com/DictService/DictService.asmx/Define?word=" + word);
Document doc = null;
DocumentBuilderFactory dbf =
DocumentBuilderFactory.newInstance();
DocumentBuilder db;
try {
db = dbf.newDocumentBuilder();
doc = db.parse(in);
} catch (ParserConfigurationException e) {
// TODO Auto-generated catch block
e.printStackTrace();
} catch (Exception e) {
// TODO Auto-generated catch block
```

```
e.printStackTrace();
}
doc.getDocumentElement().normalize();
//---retrieve all the <Definition> nodes---
NodeList itemNodes =
doc.getElementsByTagName("Definition");
String strDefinition = "";
for (int i = 0; i < definitionElements.getLength(); i++) {

Node itemNode = definitionElements.item(i);
if (itemNode.getNodeType() == Node.ELEMENT_NODE)
{
//---convert the Node into an Element---
Element definitionElement = (Element) itemNode;
//---get all the <WordDefinition> elements under
// the <Definition> element---
NodeList wordDefinitionElements =
(definitionElement).getElementsByTagName(
"WordDefinition");
strDefinition = "";
for (int j = 0; j < wordDefinitionElements.getLength(); j++) {
//---convert a <WordDefinition> Node into an Element---
Element wordDefinitionElement =
(Element) wordDefinitionElements.item(j);
//---get all the child nodes under the
// <WordDefinition> element---
NodeList textNodes =
((Node) wordDefinitionElement).getChildNodes();
strDefinition +=
((Node) textNodes.item(0)).getNodeValue() + ". ";
}
//---display the title---
Toast.makeText(getBaseContext(),strDefinition,
Toast.LENGTH_SHORT).show();
}
}
} catch (IOException e1) {
Toast.makeText(this, e1.getLocalizedMessage(),
Toast.LENGTH_LONG).show();
e1.printStackTrace();
}
}
```

The WordDefinition() method first opens an HTTP connection to the Web service, passing in the word that you are interested in:
It then uses the DocumentBuilderFactory and DocumentBuilder objects to obtain a Document (DOM) object from an XML file (which is the XML result returned by the Web service):

Once the Document object is obtained, you will find all the elements with the <Definition> tag:
As the definition of a word is contained within the <WordDefinition> element, you then proceed to extract all the definitions:
The above loops through all the <Definition> elements and then for each <Definition> element it looks for a child element named <WordDefinition>. The text content of the <WordDefinition> element contains the definition of a word. The Toast class displays each word definition that is retrieved.

Q9) Write a program for downloading binary data in networking

```
<?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"
android:layout_height="fill_parent"
>
<ImageView
android:id="@+id/img"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center" />
</LinearLayout>
```

Add the following statements to the MainActivity.java file:

```
public class MainActivity extends Activity
{
ImageView img;
private InputStream OpenHttpConnection(String urlString)
throws IOException
{
//the code used in previous section
}
private Bitmap DownloadImage(String URL)
{
Bitmap bitmap = null; InputStream in = null; try
{
in = OpenHttpConnection(URL);
bitmap = BitmapFactory.decodeStream(in); in.close();
} catch (IOException e1)
{
Toast.makeText(this, e1.getLocalizedMessage(),
Toast.LENGTH_LONG).show(); e1.printStackTrace();
}
return bitmap;
}
/** Called when the activity is first created. */ @Override
public void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState); setContentView(R.layout.main);
//---download an image---
Bitmap bitmap = DownloadImage( "http://www.streetcar.org/mim/cable/images/cable-01.jpg");
img = (ImageView) findViewById(R.id.img);
img.setImageBitmap(bitmap);
}
}
```

Q10) Develop a mobile application that uses GPS location information

Code for Activity_main.xml:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context="com.example.ex_no_5.MainActivity" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="114dp"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium"
        tools:ignore="HardcodedText" />
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/textView1"
        android:layout_marginTop="51dp"
        android:text=""
        android:textAppearance="?android:attr/textAppearanceMedium"
        tools:ignore="HardcodedText" />
    <TextView
        android:id="@+id/textView3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="47dp"
        android:text="Current Location"
        android:textAppearance="?android:attr/textAppearanceLarge"
        tools:ignore="HardcodedText" />
</RelativeLayout>
```

Code for AndroidManifest.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.ashwini.program7">
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
```

```
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
</application>

</manifest>
```

Code for MainActivity.java:

```
package com.example.vikash.program7;

import android.support.v7.app.AppCompatActivity;
import android.content.Context;
import android.location.Criteria;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.TextView;
import android.widget.Toast;
public class MainActivity extends AppCompatActivity implements LocationListener{
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        LocationManager lm=(LocationManager)getSystemService(Context.LOCATION_SERVICE);
        Criteria c=new Criteria();
        String s=lm.getBestProvider(c, false);
        if(s!=null && !s.equals(""))
        {
            Location l=lm.getLastKnownLocation(s);
            lm.requestLocationUpdates(s, 20000, 1, this);
            if(l!=null)
                onLocationChanged(l);
            else
                Toast.makeText(getApplicationContext(), "Location can't be
                        retrieved !!!", Toast.LENGTH_LONG).show();
        }
        else
            Toast.makeText(getApplicationContext(), "Provider not found !!!",
                    Toast.LENGTH_LONG).show();
    }
    @Override
    public void onLocationChanged(Location arg0) {
// TODO Auto-generated method stub
        TextView t1=(TextView)findViewById(R.id.textView1);
        t1.setText("Latitude : \n"+arg0.getLatitude());
        TextView t2=(TextView)findViewById(R.id.textView2);
        t2.setText("Longitude : \n"+arg0.getLongitude());
    }
    @Override
    public void onProviderDisabled(String arg0) {
// TODO Auto-generated method stub
```

```
    }
    @Override
    public void onProviderEnabled(String arg0) {
// TODO Auto-generated method stub
    }
    @Override
    public void onStatusChanged(String arg0, int arg1, Bundle arg2) {
// TODO Auto-generated method stub
    }
}
```