

1) What is Sensor? Explain its types?

- ⇒ Most Android-powered devices have built-in sensors that measure motion, orientation and various environmental conditions. These sensors are capable of providing raw data with high accuracy and precision, and are useful if we want to monitor three-dimensional device movement or positioning, or we want to monitor changes in the ambient environment near a device.

The Android platform supports following three broad categories of sensors:-

1. Motion Sensors:-

These sensors measure acceleration forces and rotational forces along three axes.

This category includes accelerometers, gravity sensors, gyroscopes and rotational vector sensors.

2. Environmental sensors:-

These sensors measure various environmental parameters, such as ambient air temperature and pressure, illumination and humidity.

3. Position Sensors:-

These sensors measure the physical

position of a device.

This category includes orientation sensors and magnetometers.

List of sensor types supported by Android platform :-

1) TYPE ACCELEROMETER :-

Measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y and z), including the force of gravity.

Uses :- Motion detection (Shake, tilt, etc.)

2) TYPE AMBIENT TEMPERATURE :-

Measures the ambient room temperature in degrees Celsius ($^{\circ}C$).

Uses :- Monitoring air temperatures.

3) TYPE GRAVITY :-

Measures the force of gravity in m/s^2 that is applied to a device on all three physical axes (x, y, z).

Uses :- Motion detection (Shake, tilt etc.)

4) TYPE GYROSCOPE :-

Measures a device's rate of rotation in rad/s around each of the three physical axes (x, y and z).

Uses :- Rotation detection (spin, turn, etc.)

5) TYPE - LIGHT :-

measures the ambient light level in lx.
Uses :- Controlling screen brightness.

6) TYPE - LINEAR ACCELERATION :-

measures the acceleration force in m/s^2 that is applied to a device on all three physical axes (x, y and z).

Uses :- Monitoring acceleration along a single axis.

7) TYPE - MAGNETIC FIELD :-

measures the ambient geomagnetic field for all three physical axes (x, y, z) in μT .

Uses :- Creating a Compass.

8) TYPE - ORIENTATION :-

Measures degrees of rotation that a device makes around all three physical axes (x, y, z).

Uses :- Determining device position.

9) TYPE - PRESSURE :-

measures the ambient air pressure in hPa or mbar.

Uses :- monitoring air pressure changes.

10) TYPE - PROXIMITY :-

measures the proximity of an object in cm relative to the view screen of a device.

Uses :- Phone position during a call.

11) TYPE_RELATIVE_HUMIDITY :-
Measures the relative ambient humidity in percent (%).
Uses :- Monitoring dewpoint, absolute and relative humidity.

12) TYPE_ROTATION_VECTOR :-
Measures the orientation of a device by providing the three elements of the device's rotation vector.
Uses :- Motion detection and rotation detection.

13) TYPE_TEMPERATURE :-
Measures the temperature of the device in degrees Celsius ($^{\circ}\text{C}$).
Uses :- Monitoring temperatures.

2) Explain Sensor Framework.

⇒ - We can access sensors and acquire raw sensor data by using the Android sensor framework.

The sensor framework is part of the android hardware package and includes the following classes and interfaces:

1) SensorManager :-

- We can use this class to create an instance of the sensor service.

This class provides various methods for accessing and listing sensors, registering and unregistering sensor.

event listeners, and acquiring orientation information. This class also provides several sensor constants that are used to report sensor accuracy, set data acquisition rates and calibrate sensors.

2) Sensor :-

We can use this class to create an instance of a specific sensor. This class provides various methods that let determine a sensor's capabilities.

3) SensorEvent :-

The system uses this class to create a sensor event object, which provides information about a sensor event.

4) SensorEventListener :-

We can use this interface to create two callback methods that receive notifications, when sensor values change or when sensor accuracy changes.

3) Explain AsyncTask with example. or
Explain methods used in android AsyncTask class.

⇒

- Android AsyncTask is an abstract category provided by android which provides United States the freedom to perform significant tasks within the background and keep the UI thread lightweight thus making the application more responsive.
- Android application runs on one thread once launched. because of ~~the~~ this single thread model tasks that take longer time to fetch the response can make the application non-responsive.
- To avoid this we have a tendency to use Android AsyncTask to perform the significant tasks in background on a dedicated thread and passing the results back to the UI thread.

Hence use of AsyncTask in android application keeps the UI thread responsive at all times.

Methods used in AsyncTask class are:-

1) doInBackground() :-

This method contains the code which needs to be executed in background. In this method we can send results multiple times to the UI thread by

publishProgress() method.

To notify that the background processing has been completed we just need to use the return statements.

2) onPreExecute() :-

This method contains the code which is executed before the background processing starts.

3) onPostExecute() :-

This method is called after doInBackground method completes processing. Result from doInBackground is passed to this method.

4) onProgressUpdate() :-

This method receives progress updates from doInBackground method, which is published via publishProgress method and this method can use this progress update to update the UI thread.

The three generic types used in an android AsyncTask class are :-

- 1) Params :- The type of the parameters sent to the task upon execution.
- 2) Progress :- The type of the progress units published during the background computation.
- 3) Result :- The type of the result of the background computation.

- 4) Describe the term Audio Capture in detail.
- ⇒ - The Android multimedia system framework includes support for capturing and encoding variety of common audio formats, so we will simply integrate audio into our applications. We can record audio using the `MediaRecorder` APIs if supported by the device hardware.

Performing Audio Capture :-

- 1) Create a new instance of `android.media.MediaRecorder`.
- 2) Set the audio source using `MediaRecorder.setAudioSource()`.
We will probably want to use `MediaRecorder.AudioSource.MIC`.
- 3) Set output file format using `MediaRecorder.setOutputFormat()`.
- 4) Set output file name using `MediaRecorder.setOutputFile()`.
- 5) Set the audio encoder using `MediaRecorder.setAudioEncoder()`.
- 6) Call `MediaRecorder.prepare()` on the `MediaRecorder` instance.
- 7) To start audio capture, call `MediaRecorder.start()`.
- 8) To stop audio capture, call `MediaRecorder.stop()`.
- 9) When we are done with the `MediaRecorder` instance, call `MediaRecorder.release()` on it. Calling `MediaRecorder.release()` is always recommended to free the resources.

5) Explain Camera in Android in detail.
There are two ways to integrate the camera module.

1) Using In-built Camera App:-

The easiest way to integrate Camera feature is, use the in-built camera app that comes with the device. To launch the camera app and it'll come back the information we simply need to pass the acceptable intent with necessary flags.

Rest are taken care by the camera app and it will come back the information to our app. The disadvantage of this methodology is, we can not customize the camera interface as we would be launching third party camera app.

2) Writing Custom Camera App:-

Building our own camera module takes some effort as we would like to create everything from scratch starting with user interface to integrating camera API.

Basics for Camera:-

1) camera :- This class is the primary API for controlling device cameras. This class is used to take picture or videos when we are building a camera application.

2) Surface View :-

This class is used to present a live camera preview to the user.

3) MediaRecorder :-

This class is used to record video from the camera.

4) Intent :-

An intent action type of Mediastore.

ACTION_IMAGE_CAPTURE or Mediastore.

ACTION_VIDEO_CAPTURE can be used to capture images or videos without directly using the camera object.

- Before starting development on our application with the camera API, we should make sure our manifest has the appropriate declarations to allow use of camera hardware and other related features.

1) Camera Permission :-

Our application must request permission to use a device camera.

```
<uses-permission android:name="android.permission.CAMERA"/>
```

2) Camera Features :-

Our application must also declare use of camera feature. For example:

```
<uses-feature android:name="android.hardware.camera"/>
```

3) Storage Permission :-

If our application saves images or videos to the device's external storage, we must also specify this in the manifest.

```
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```


4) Audio Recording Permission :-

For recording audio with video capture, our application must request the audio capture permission.

```
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
```

5) Location Permission :-

If our application tags images with GPS location information, we must request location permission:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
```

Using Existing Camera Apps :-

- A quick way to enable taking pictures or videos in our application without a lot of extra code is to use an intent to invoke an existing Android camera application.

- A camera intent make a request to capture a picture or video clip through an existing camera app and then returns control back to our application.

The procedure for invoking a camera intent follows these general steps :-

1) Compose a Camera Intent :-

Create an Intent that requests an image or video, using one of these intent types :-

i) MediaStore.ACTION_IMAGE_CAPTURE :-

Intent action type for requesting an image.

from an existing camera application
ii) MediaStore ACTION_VIDEO_CAPTURE :-
Intent action type for requesting a video
from an existing camera application.

2) Start the Camera Intent :- Use the
startActivityForResult() method to execute
the camera intent.

After we start the intent, the camera
application user interface appears on
the device screen and the user can
take a picture or video.

3) Receive the Intent Result :-

Set up an onActivityResult() method
in our application to receive the callback
and ~~at~~ data from the camera intent.
When the user finishes taking a picture
or video (or cancel the operation), the
system calls this method.

(Program as discuss in practical)

6* Explain Bluetooth in Android in detail.

- Android platform includes support for the Bluetooth framework that allows a device to wirelessly exchange data with other Bluetooth devices.
- In android, ~~Blue~~ Bluetooth is a communication network protocol, which allow a devices to connect wirelessly to exchange the data with other Bluetooth devices.
- By using android Bluetooth API's in android applications, we can perform following functionalities.
 - 1) Scan for the available Bluetooth devices within the range.
 - 2) Use local Bluetooth adapter for paired Bluetooth devices.
 - 3) Connect to other devices through service discovery.
 - 4) Transfer data to and from other devices.
 - 5) Manage multiple connections.
- To transfer the data between two Bluetooth devices, first they must establish a communication channel using pairing process. The devices which we are going to pair must be discoverable and should accept the incoming connection requests. Once the device accepts the pairing request the two devices will exchange a security keys to complete the bounding process and the devices will catch these security keys for later use.

- Once the pairing and bonding process completes, the devices are ready to exchange the required information. When the session is complete, the device that initiated the pairing request will release the channel that linked to the discoverable device.
- The two devices remain bounded, so they can reconnect automatically during a future session as long as they are in the range of each other.

(Program discuss in Practical)

1) What is Animation. Enlist its types. Explain any one.

⇒

- Animation is the process of creation motion and shape change.

Android Animation in android is possible from many ways.

In this section we will discuss one easy and widely used way of making animation called tweened animation.

- Animations are useful when the screen changes state.
- Animation is the process of creating motion and shape change.
- Animations notify users about what's going on in our app and improve the mental model of our app's interface.

Camlin Page
Date / /

Android defines following three types of animations:-

1) View Animation:-

- This is the simplest animation used in Android.

It defines the properties of our views that should be animated using a technique called Tween Animation.

- It takes the following parameters i.e. size, time duration, rotation angle, start value, end value and perform the required animation on that object.

We can execute the animation by specifying transformations on our view. This can be done in XML resource files or programmatically.

- Android View animation can make animation on any view objects, such as ImageView, TextView or Button objects.

View animation can only animate a simple properties like position, size, rotation and the alpha property that allows us animate the transparency of a view.

- The drawback of this mechanism is that it can only be applied to views.

2) Property Animation:-

- This animation was introduced in Android 3.0.

It allows the user to animate anything.

- Property animations are highly customizable, we can specify the duration, the number of repeats, the type of interpolation and the

frame rate of the animation.

The property Animation system is always preferred for more complex animations.

- Property animations allow us to animate any property of any object from one value to another over a specified duration.
- We are not limited by predefined animation types or by the type of object that we want to animate.
- At first the usage of the property animation system might seem a little complicated.
- Common properties commonly animated on views include:

1) alpha :- Fade in or out

2) rotation, rotationX, rotationY :-
Spin or Flip

3) scaleX, scaleY :- Grow or shrink

4) x, y, z :- Position

5) translationX, translationY, translationZ :-
offset from position.

3) Drawable Animation :-

- This animation allows the user to load drawable resources and display them one frame after another.

This method of animation is useful when user wants to animate things that are easier to represent with Drawable resources.

- Curriculum Page
Date / /
- To use Animations in Android, Android has provided as a class called Animation.
 - The Animation class has many methods given below:

- 1) start() :- This method will start the animation.
- 2) setDuration(long duration) :- This method sets the duration of an animation.
- 3) getDuration() :- This method ends the animation. gets the duration.
- 4) end() :- This method ends the animation.
- 5) cancel() :- This method cancels the animation.

(Program as discuss in practical)