

Experiment no. 10

Aim: To implement GPS application in Android.

Requirements: Compatible version of android studio.

Theory:

Request location permissions

To protect user privacy, apps that use location services must request location permissions.

When you request location permissions, follow the same best practices as you would for any other runtime permission. One important difference when it comes to location permissions is that the system includes multiple permissions related to location. Which permissions you request, and how you request them, depend on the location requirements for your app's use case.

This page describes the different types of location requirements and provides guidance on how to request location permissions in each case.

Types of location access

Each permission has a combination of the following characteristics:

Category: Either foreground location or background location.

Accuracy: Either precise location or approximate location.

Foreground location

If your app contains a feature that shares or receives location information only once, or for a defined amount of time, then that feature requires foreground location access. Some examples include the following:

Within a navigation app, a feature allows users to get turn-by-turn directions.

Within a messaging app, a feature allows users to share their current location with another user.

The system considers your app to be using foreground location if a feature of your app accesses the device's current location in one of the following situations:

An activity that belongs to your app is visible.

Your app is running a foreground service. When a foreground service is running, the system raises user awareness by showing a persistent notification. Your app retains access when it's placed in the background, such as when the user presses the Home button on their device or turns their device's display off.

Additionally, it's recommended that you declare a foreground service type of location, as shown in the following code snippet. On Android 10 (API level 29) and higher, you must declare this foreground service type.

```
<!-- Recommended for Android 9 (API level 28) and lower. -->
```

```
<!-- Required for Android 10 (API level 29) and higher. -->
```

```
<service
```

```
    android:name="MyNavigationService"
```

```
    android:foregroundServiceType="location" ... >
```

```
<!-- Any inner elements would go here. -->
```

```
</service>
```

You declare a need for foreground location when your app requests either the `ACCESS_COARSE_LOCATION` permission or the `ACCESS_FINE_LOCATION` permission, as shown in the following snippet:

```
<manifest ... >
```

```
<!-- Always include this permission -->
```

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

```
<!-- Include only if your app benefits from precise location access. -->
```

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

```
</manifest>
```

Background location

An app requires background location access if a feature within the app constantly shares location with other users or uses the Geofencing API. Several examples include the following:

Within a family location sharing app, a feature allows users to continuously share location with family members.

Within an IoT app, a feature allows users to configure their home devices such that they turn off when the user leaves their home and turn back on when the user returns home.

The system considers your app to be using background location if it accesses the device's current location in any situation other than the ones described in the foreground location section. The background location precision is the same as the foreground location precision, which depends on the location permissions that your app declares.

On Android 10 (API level 29) and higher, you must declare the `ACCESS_BACKGROUND_LOCATION` permission in your app's manifest in order to request background location access at runtime. On earlier versions of Android, when your app receives foreground location access, it automatically receives background location access as well.

```
<manifest ... >
```

```
<!-- Required only when requesting background location access on
```

```
Android 10 (API level 29) and higher. -->
```

```
<uses-permission
```

```
android:name="android.permission.ACCESS_BACKGROUND_LOCATION" />
```

```
</manifest>
```

Accuracy

Android supports the following levels of location accuracy:

Approximate

Provides a device location estimate. If this location estimate is from the `LocationManagerService` or `FusedLocationProvider`, this estimate is accurate to within about 3 square kilometers (about 1.2 square miles). Your app can receive locations at this level of accuracy when you declare the `ACCESS_COARSE_LOCATION` permission but not the `ACCESS_FINE_LOCATION` permission.

Precise

Provides a device location estimate that is as accurate as possible. If the location estimate is from `LocationManagerService` or `FusedLocationProvider`, this estimate is usually within about 50 meters (160 feet) and is sometimes as accurate as within a few meters (10 feet) or better. Your app can receive locations at this level of accuracy when you declare the `ACCESS_FINE_LOCATION` permission.

If the user grants the approximate location permission, your app only has access to approximate location, regardless of which location permissions your app declares.

Your app should still work when the user grants only approximate location access. If a feature in your app absolutely requires access to precise location using the ACCESS_FINE_LOCATION permission, you can ask the user to allow your app to access precise location.

Code:

```
package com.yayandroid.locationmanager.sample;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;

import com.yayandroid.locationmanager.sample.activity.SampleActivity;
import com.yayandroid.locationmanager.sample.fragment.SampleFragmentActivity;
import com.yayandroid.locationmanager.sample.service.SampleServiceActivity;

public class MainActivity extends AppCompatActivity {

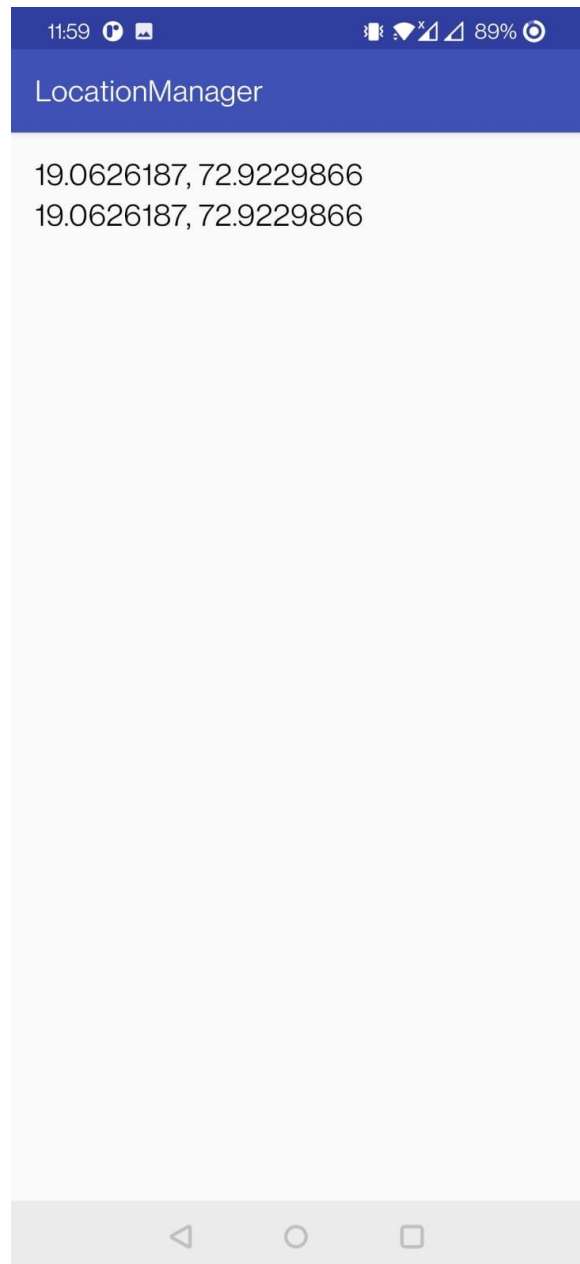
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

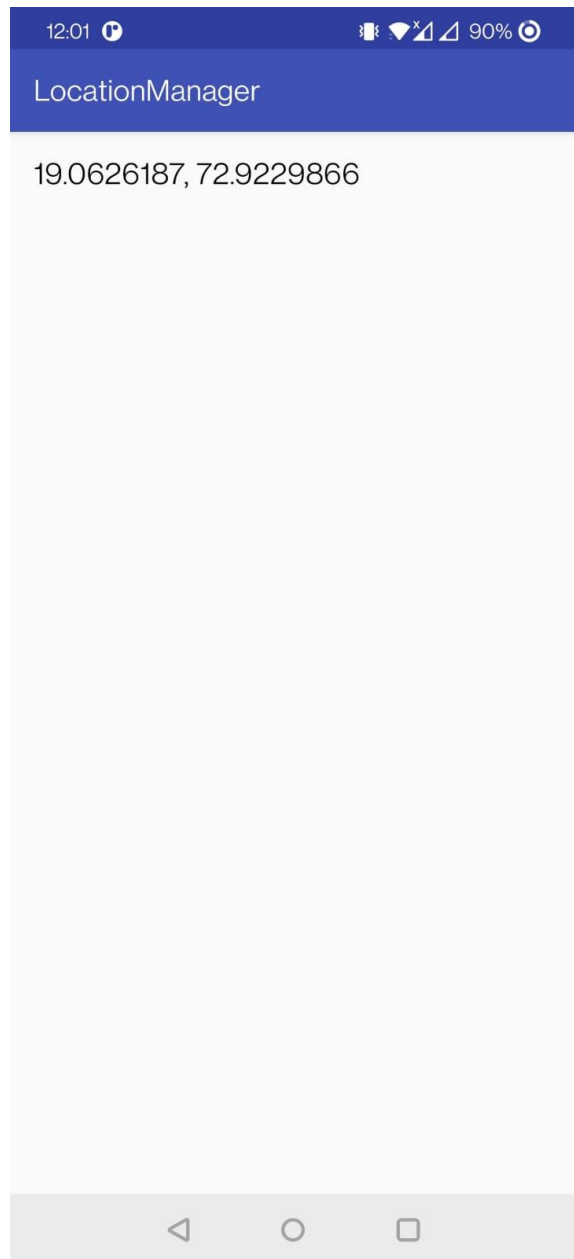
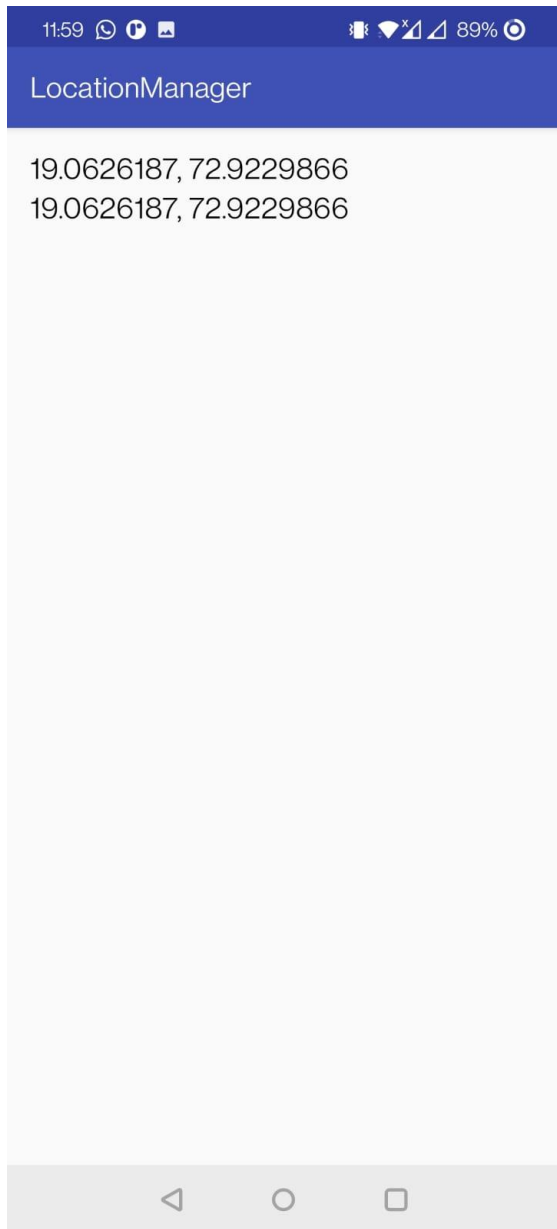
    public void inActivityClick(View view) {
        startActivity(new Intent(this, SampleActivity.class));
    }

    public void inFragmentClick(View view) {
        startActivity(new Intent(this, SampleFragmentActivity.class));
    }

    public void inServiceClick(View view) {
        startActivity(new Intent(this, SampleServiceActivity.class));
    }
}
```

Output:





Conclusion: We have successfully implemented Android GPS application using Android Studio.