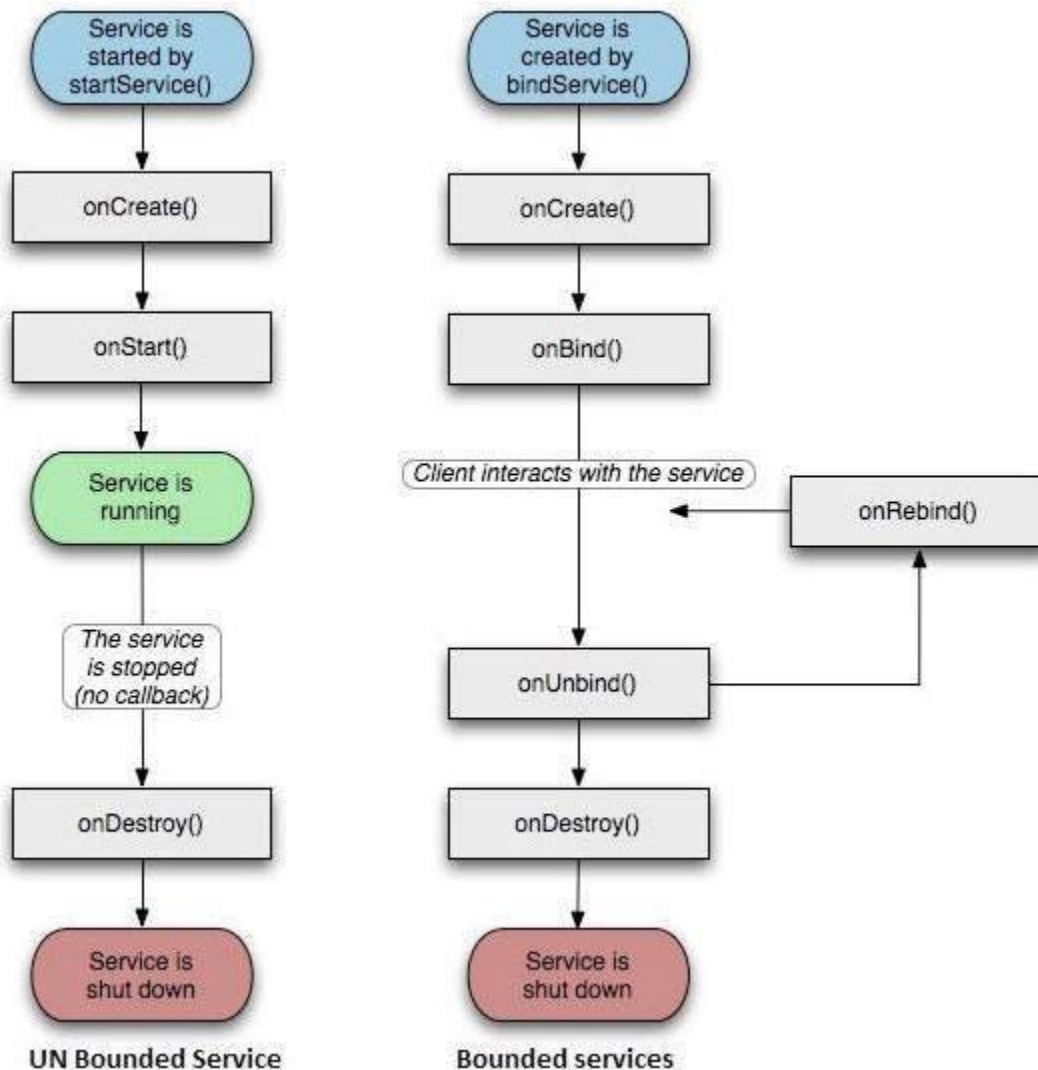


## Practical related question:-

1. Draw the lifecycle of service.



2. Differentiate between bounded services and unbounded service.

unbounded	bounded
Unbounded Service is used to perform long repetitive task	Bounded Service is used to perform background task in bound with another component
Unbound Service gets starts by calling <b>startService()</b> .	bounded Service gets starts by calling <b>bindService()</b> .

Unbound Service is stopped or destroyed explicitly by calling <b>stopService()</b> .	bounded Service is unbind or destroyed by calling <b>unbindService()</b> .
Unbound Service is independent of the component in which it is started.	bound Service depends on the component in which it is started.

### 3. Describe startService() and bindService() method.

**Startservice:-** Creating a **started service**. A **started service** is one that another component **starts** by calling startService() , which results in a call to the **service's** onStartCommand() method. When a **service** is **started**, it has a lifecycle that's independent of the component that **started** it.

**bindService():-** allows components (such as activities) to **bind** to the **service**, send requests, receive responses, and perform interprocess communication (IPC). A bound **service** typically lives only while it serves another application component and does not run in the background indefinitely

## Exercise:-

### MainActivity.java

```
package com.example.practical20;

import androidx.appcompat.app.AppCompatActivity;

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

public class MainActivity extends AppCompatActivity {
    Button wifi,service_on,service_off;
    Intent serviceeg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        wifi=(Button) findViewById(R.id.wifi);
        service_on=(Button) findViewById(R.id.service_on);

        service_off=(Button) findViewById(R.id.service_off);

        wifi.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent wifion=new Intent(MainActivity.this,Wifion.class);
                startService(wifion);
            }
        });

        service_on.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                serviceeg=new Intent(MainActivity.this,Serviceeg.class);
                startService(serviceeg);
            }
        });

        service_off.setOnClickListener(new View.OnClickListener() {
```

```

        @Override
        public void onClick(View v) {
            stopService(serviceeg);
        }
    });
}
}

```

## Serviceeg.java

```

package com.example.practical20;

import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.net.wifi.WifiManager;
import android.os.IBinder;
import android.widget.Toast;

import androidx.annotation.Nullable;

public class Serviceeg extends Service {
    @Nullable
    @Override

    public int onStartCommand(Intent intent, int flags, int startId) {
        Toast.makeText(this, "Service Started", Toast.LENGTH_SHORT).show();
        return START_STICKY;
    }

    public void onDestroy() {
        Toast.makeText(this, "Service Stopped", Toast.LENGTH_SHORT).show();
    }

    public IBinder onBind(Intent intent) {
        return null;
    }
}

```

## Wifion.java

```
package com.example.practical20;

import android.annotation.SuppressLint;
import android.app.Service;
import android.content.Context;
import android.content.Intent;
import android.net.wifi.WifiManager;
import android.os.IBinder;
import android.widget.Toast;

import androidx.annotation.Nullable;

public class Wifion extends Service {

    @Nullable
    @Override
    @SuppressLint("ServiceCast")
    public int onStartCommand(Intent intent, int flags, int startId) {
        Toast.makeText(this, "Starting Wifi on", Toast.LENGTH_SHORT).show();

        WifiManager wifi=(WifiManager) getApplicationContext().getSystemService(C
ontext.WIFI_SERVICE);
        wifi.setWifiEnabled(true);

        return START_STICKY;
    }
    public IBinder onBind(Intent intent) {
        return null;
    }
}
```

## Activity\_Main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.
android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/service_off"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="136dp"
        android:layout_marginTop="480dp"
        android:text="Stop Service"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/service_on"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="128dp"
        android:layout_marginTop="420dp"
        android:text="Start Service"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/wifi"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="136dp"
        android:layout_marginTop="184dp"
        android:text="Turn on Wifi"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```