```
package com.internshala.echo.activities
import android.os.Bundle
import android.support.v4.widget.DrawerLayout
import android.support.v7.app.ActionBarDrawerToggle
import android.support.v7.app.AppCompatActivity
import android.support.v7.widget.DefaultItemAnimator
import android.support.v7.widget.LinearLayoutManager
import android.support.v7.widget.RecyclerView
import android.support.v7.widget.Toolbar
import com.internshala.echo.fragments.MainScreenFragment
import com.internshala.echo.R
import com.internshala.echo.adapters.NavigationDrawerAdapter

class MainActivity : AppCompatActivity() {

/*The list for storing the names of the items in list of navigation drawer*/
     var navigationDrawerIconsList: ArrayList<String> = arrayListOf()

/*Images which will be used inside navigation drawer*/
     var images_for_navdrawer = intArrayOf(R.drawable.navigation_allsongs,
R.drawable.navigation_favorites, R.drawable.navigation_settings,
R.drawable.navigation_aboutus)

/*We made the drawer layout as an object of this class so that this object can also be used
as same inside the adapter class without any change in its value*/
     object Statified {
          var drawerLayout: DrawerLayout? = null
     }

override fun onCreate(savedInstanceState: Bundle?) {
          super.onCreate(savedInstanceState)
          setContentView(R.layout.activity_main)
          val toolbar = findViewById<Toolbar>(R.id.toolbar)
          setSupportActionBar(toolbar)

     /*This syntax is used to access the objects inside the class*/
          MainActivity.Statified.drawerLayout = findViewById(R.id.drawer_layout)

     /*Adding names of the titles using the add() function of ArrayList*/
          navigationDrawerIconsList.add("All Songs")
          navigationDrawerIconsList.add("Favorites")
          navigationDrawerIconsList.add("settings")
          navigationDrawerIconsList.add("About Us")

     val toggle = ActionBarDrawerToggle(this@MainActivity,
MainActivity.Statified.drawerLayout, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close)
          MainActivity.Statified.drawerLayout?.addDrawerListener(toggle)
          toggle.syncState()
          val mainScreenFragment = MainScreenFragment()
          this.supportFragmentManager
                    .beginTransaction()
                    .add(R.id.details_fragment, mainScreenFragment, "MainScreenFragment")
                    .commit()

     /*Now we create a variable of Navigation Drawer adapter and initialise it with the
params required. As you remember that while creating a class for the navigation drawer
adapter,
          * we gave it some params which are required for initialising the class. These
params are the list, images and the context for the adapter file respectively*/
          val _navigationAdapter = NavigationDrawerAdapter(navigationDrawerIconsList,
images_for_navdrawer, this)

/*Here the function notifyDataSetChanged() tells the adapter that the data you were holding
has been changed and thus you should refresh the list*/
```

```
        _navigationAdapter.notifyDataSetChanged()

/*Declaring the variable navigation_recycler_view for the list inside the navigation
drawer*/
        val navigation_recycler_view =
findViewById<RecyclerView>(R.id.navigation_recycler_view)

/*Here we set a layout manager which aligns the items in a recycler view. As we want to set
the items vertically one below the other we use a linear layout manager.*/
        navigation_recycler_view.layoutManager = LinearLayoutManager(this)

/*As the name is suggesting the item animator is used to animate the way the items appear in
a recycler view. As we used the default item animator, here we will just see the items
         * appear as they come without any transition */
        navigation_recycler_view.itemAnimator = DefaultItemAnimator()

/*Now we set the adapter to our recycler view to the adapter we created*/
        navigation_recycler_view.adapter = _navigationAdapter

/*As the code setHasFixedSize() suggests, the number of items present in the recycler view
are fixed and won't change any time*/
        navigation_recycler_view.setHasFixedSize(true)
    }

override fun onStart() {
        super.onStart()
    }
}
```