```
package com.internshala.echo.fragments
import android.app.Activity
import android.content.Context
import android.os.Bundle
import android.support.v4.app.Fragment
import android.view.LayoutInflater
import android.view.Menu
import android.view.View
import android.view.ViewGroup
import android.widget.Switch
import com.internshala.echo.R
/**
 * A simple [Fragment] subclass.
 */
/*The settings fragment class is used for handling the events inside the settings fragment*/

class SettingsFragment : Fragment() {

    var myActivity: Activity? = null

    /*Declaring the switch used*/
    var shakeSwitch: Switch? = null

    /*Here the change in switch will lead to turning on and off of a setting so we need to
persist the changes
    * This will be done with the help of Shared preferences*/
    object Statified {
        var MY_PREFS_NAME = "ShakeFeature"
    }

    override fun onCreateView(inflater: LayoutInflater?, container: ViewGroup?,
                              savedInstanceState: Bundle?): View? {
        // Inflate the layout for this fragment
        val view = inflater!!.inflate(R.layout.fragment_settings, container, false)

        /*Linking switch to its view*/
        shakeSwitch = view?.findViewById(R.id.switchShake)
        return view
    }

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
    }

    override fun onAttach(context: Context?) {
        super.onAttach(context)
        myActivity = context as Activity
    }

    override fun onAttach(activity: Activity?) {
        super.onAttach(activity)
        myActivity = activity
    }

    override fun onActivityCreated(savedInstanceState: Bundle?) {
        super.onActivityCreated(savedInstanceState)
        val prefs = myActivity?.getSharedPreferences(Statified.MY_PREFS_NAME,
Context.MODE_PRIVATE)

        var isAllowed = prefs?.getBoolean("feature", false)
```

```
        /*Checking the value of the feature as to whether it is ON or OFF*/
        if (isAllowed as Boolean) {

            /*If feature is ON then we make the switch to checked*/
            shakeSwitch?.isChecked = true
        } else {

            /*Else unchecked*/
            shakeSwitch?.isChecked = false
        }

        /*Now we handle the change events i.e. when the switched is turned ON or OFF*/
        shakeSwitch?.setOnCheckedChangeListener({ compoundButton, b ->
            if (b) {

                /*If the switch is turned on we then make the feature to be true*/
                val editor = myActivity?.getSharedPreferences(Statified.MY_PREFS_NAME,
Context.MODE_PRIVATE)?.edit()
                editor?.putBoolean("feature", true)
                editor?.apply()
            } else {

                /*Else the feature remains false*/
                val editor = myActivity?.getSharedPreferences(Statified.MY_PREFS_NAME,
Context.MODE_PRIVATE)?.edit()
                editor?.putBoolean("feature", false)
                editor?.apply()
            }
        })
    }

    override fun onPrepareOptionsMenu(menu: Menu?) {
        super.onPrepareOptionsMenu(menu)
    }
}
```