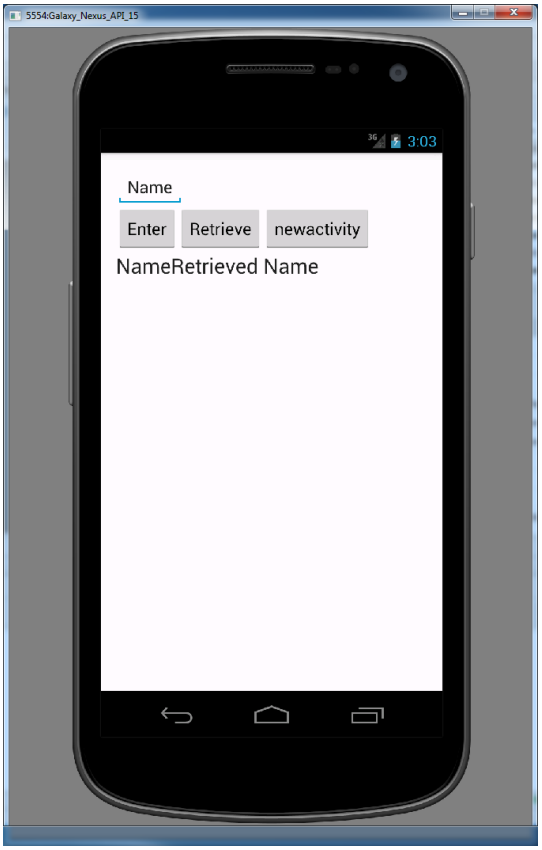
Lab 5: Saving Data with Shared Preferences

Using shared preferences is a means of saving a small number of name-value data items that you wish to keep stored on the user's device. A SharedPreferences object points to a file containing name-value pairs and provides simple methods to read and write them. The file is available to all activities within the java package.

The following example shows how to take some input, store it as a shared preference and then retrieve that data again. The shared preference can be accessed by different activities within the app. Start by setting up a new empty project with the following layout.



The layout file is provided at the end of this sheet. There is one input EditText box, three buttons and two output TextViews.

The user will enter their name – the Enter button will echo their name to one of the TextViews and also, silently, use stored preferences to save that information on the memory of the device. The retrieve button will obtain the name from the stored preference and display it in the other textView.

The third button, not initially implemented, will open up another activity and display the name there.

The MainActivity.java file is also included at the end of this sheet. Copy and paste this into your project's MainActivity. Let's have a look at this first.

Have a look at how the onClick() method is set to deal with the any of the three buttons being clicked. This should be familiar to you.

You should now note how the shared preferences are set up. The following lines are all required – identify where they are placed in the activity

```
SharedPreferences mySharedPreferences;    // declares a shared preference object

// attaches the shared preference object to a file (myPrefsFile) stored on the device memory in the space set aside for the app
mySharedPreferences = getSharedPreferences("myPrefsFile", MainActivity.MODE_PRIVATE);

SharedPreferences.Editor editor = mySharedPreferences.edit();  // sets up an editor

editor.putString("clientName", name);       //  adds a name value pair to the editor
editor.apply();                             // takes the editor contents and puts into the stored file
```

The last two lines are in the "to do" action when the Enter button is pressed.

The “to do” action for the retrieve button gets the information from the shared preference and displays it in a textView.

```
String nameSF = mySharedPreferences.getString("clientName", "not something");
storedName.setText(nameSF);
```

There seems quite a lot of work for little gain here – but one thing you should now be able to do is modify the MainActivity so that when it starts up for a second time the name already entered is used. This is much more user friendly.

The third button links to a new activity that is not implemented yet. Implement it now. The code you need in the new activity is:

```
// need to connect to the object "myPrefsFile"
SharedPreferences mySharedPreferences = getSharedPreferences("myPrefsFile",
MainActivity.MODE_PRIVATE);

String name = mySharedPreferences.getString("clientName", "no name entered");

Toast.makeText(this, name , Toast.LENGTH_LONG).show();
TextView storedName = (TextView)findViewById(R.id.textName);
storedName.setText(name);
```

Here shared preferences have been used to do the sort of thing that putExtra did last week. The advantage of using shared preferences is that the information is stored and can be used by any activity – not just transferred from one activity to another. This could be useful for you assessment as explained in the lecture.

The next thing you could try is dealing with types of data other than string. There are a series of put statements as shown below:

```
editor.putBoolean("isTrue", true);  // a series of putSomething statements that writes the data
editor.putFloat("lastFloat", 1.5f);   // in name (within the quotes) value pairs
editor.putInt("wholeNumber", 2);
editor.putString("clientName","Robert Burns");
editor.apply();   // makes the data become stored
```

Each of the data types has its own way of being retrieved and can be converted to a string for displaying in a TextView as shown below:

```
Boolean flag = mySharedPreferences.getBoolean("isTrue",false);
float totalCost = mySharedPreferences.getFloat("lastFloat",0.0f);
int noOfAdults = mySharedPreferences.getInt("wholeNumber",0);
String name = mySharedPreferences.getString("TextEntry", "not something");

// String TextEntry = String.valueOf(flag);
// String TextEntry = String.valueOf(totalCost);
String TextEntry = String.valueOf(noOfAdults);
```

Layout File

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:orientation="vertical"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   tools:context="com.example.myapplication.MainActivity">

   <EditText
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/nameInput"
      android:layout_alignParentTop="true"
      android:layout_alignParentLeft="true"
      android:layout_alignParentStart="true"
      android:text="Name" />

   <LinearLayout
      android:orientation="horizontal"
      android:layout_width="match_parent"
      android:layout_height="wrap_content">

      <Button
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Enter"
         android:id="@+id/storeName"
         android:layout_below="@+id/nameIn"
         android:layout_alignParentLeft="true"
         android:layout_alignParentStart="true" />

      <Button
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="Retrieve"
         android:id="@+id/retrieveName" />

      <Button
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:text="newactivity"
         android:id="@+id/newActivity" />
   </LinearLayout>

   <LinearLayout
```

```
    android:orientation="horizontal"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceLarge"
      android:text="Name"
      android:id="@+id/nameIn"
      android:layout_below="@+id/storeName"
      android:layout_alignParentLeft="true"
      android:layout_alignParentStart="true" />

    <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:textAppearance="?android:attr/textAppearanceLarge"
      android:text="Retrieved Name"
      android:id="@+id/nameOut"
      android:layout_gravity="center_horizontal" />
  </LinearLayout>
</LinearLayout>
```

MainActivity.java

```
import android.app.Activity;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends Activity implements View.OnClickListener{

  EditText userName;
  Button inButton, outButton, toNewActivity;
  SharedPreferences mySharedPreferences;
  String name;
  TextView nameIn, storedName;

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

    userName = (EditText)findViewById(R.id.nameInput);
```

```
        nameIn = (TextView)findViewById(R.id.nameIn);
        storedName = (TextView)findViewById(R.id.nameOut);

        inButton = (Button)findViewById(R.id.storeName);
        inButton.setOnClickListener(this);

        outButton = (Button)findViewById(R.id.retrieveName);
        outButton.setOnClickListener(this);

        toNewActivity = (Button)findViewById(R.id.newActivity);
        toNewActivity.setOnClickListener(this);

        mySharedPreferences = getSharedPreferences("myPrefsFile", MainActivity.MODE_PRIVATE);
        // declares the shared preferences object
    }

    @Override
    public void onClick(View view) {

        SharedPreferences.Editor editor = mySharedPreferences.edit();  // creates the editor

        switch (view.getId()) {
            case R.id.storeName:
                //Toast.makeText(this, "Store name button.", Toast.LENGTH_LONG).show();
                name = userName.getText().toString();
                nameIn.setText(name);
                editor.putString("clientName", name);
                editor.apply();
                break;
            case R.id.retrieveName:
                //Toast.makeText(this, "Retrieve name button", Toast.LENGTH_LONG).show();
                String nameSF = mySharedPreferences.getString("clientName", "not something");
                storedName.setText(nameSF);
                break;
            case R.id.newActivity:
                //Toast.makeText(this, "New Activity button", Toast.LENGTH_LONG).show();
                //Intent intent = new Intent(this,ShowName.class);
                //startActivity(intent);
                break;
        }
    }
}
```