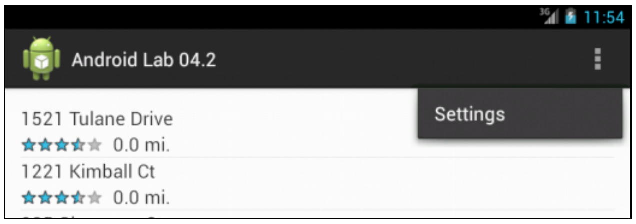
Lab 4.2 – Create a Menu Item

Overview:

In this lab you create a new menu options that can be selected by the user.

Step	Description

1.	Create new menu options for adding a new event, seeing a map of events and displaying a preferences page. The Activity wizard automatically creates a default menu with a single option. You will be modifying this default menu.
	
	The default menu is contained in the file "res/menu/menu.xml" and looks like the following:
	
	`<menu xmlns:android="http://schemas.android.com/apk/res/android" >`
	
	`    <item`
	`        android:id="@+id/action_settings"`
	`        android:orderInCategory="100"`
	`        android:showAsAction="never"`
	`        android:title="@string/action_settings"/>`
	
	`</menu>`
	
	In Android 4.1 and above the menu appears as 3 stacked dots in the upper right of the screen. The menu is activated by clicking on the dots which causes a list of menu items to appear.
	
	 11:54
	Android Lab 04.2
	Settings
	1521 Tulane Drive
	0.0 mi.
	1221 Kimball Ct
	0.0 mi.

a.	In this step you will create the XML file which defines the menu options. Note: In new versions of Android Development Toolkit this will be generated for you automatically. Create a new layout to define the menu. Start the New Android XML File wizard by selecting “AndroidLab/res/menu” and right clicking to reach the context menu Select "New -> Other -> Android -> Android XML File" and click “Next” and provide the following: Resource Type: Menu Project: AndroidLab File: main Root Element: menu

Click “Finish” and verify that the following file has been created "/res/menu/main_menu.xml"

b. Add items to the menu. The menu will have options for

- Adding a new event
- Showing the events on a map
- Displaying the preferences page

Edit the file “res/menu/menu.xml” and add “item” elements within the “menu” tag which will declare the menu options. The completed file should look like the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<menu
      xmlns:android="http://schemas.android.com/apk/res/android">
      <item
            android:id="@+id/mi_add_event"
            android:title="@string/mi_add_event" />
      <item
            android:id="@+id/mi_show_map"
            android:title="@string/mi_show_map" />
      <item
            android:id="@+id/mi_prefs"
            android:title="@string/mi_prefs" />
</menu>
```

c. The menu layout file references some external strings. Enter string values for each of the menu items.

Open the file “res/values/strings.xml” and add the following:

```
<string name="mi_add_event">Add Event</string>
<string name="mi_show_map">Map</string>
<string name="mi_prefs">Prefs</string>
```

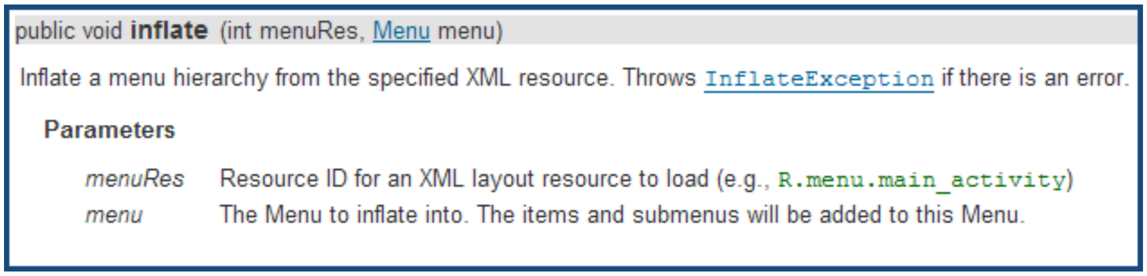
2. Modify the main activity to display the menu

a. Add (or modify) the code for declaring the menu. Open the file “src/com.garagze/MainActivity.java” and add the following method to the end of the program:

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
  getMenuInflater().inflate(R.menu.main, menu);
  return true;
}
```

Note: This method may already have been generated for you.

Review the javadoc for the “MenuInflator” class.



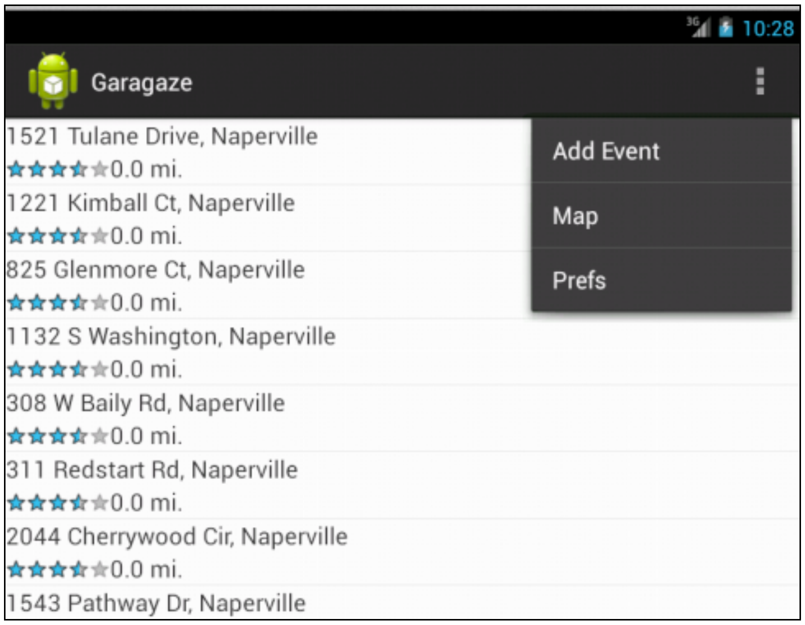
public void **inflate** (int menuRes, Menu menu)

Inflate a menu hierarchy from the specified XML resource. Throws `InflateException` if there is an error.

Parameters

menuRes	Resource ID for an XML layout resource to load (e.g., `R.menu.main_activity`)
menu	The Menu to inflate into. The items and submenus will be added to this Menu.

Test the application by running it on the emulator. Press the “Menu” icon (vertical series of 3 squares) to display the menu. The menu options should display. You can click them but no functionality has been added to them yet.



1521 Tulane Drive, Naperville
0.0 mi.
1221 Kimball Ct, Naperville
0.0 mi.
825 Glenmore Ct, Naperville
0.0 mi.
1132 S Washington, Naperville
0.0 mi.
308 W Baily Rd, Naperville
0.0 mi.
311 Redstart Rd, Naperville
0.0 mi.
2044 Cherrywood Cir, Naperville
0.0 mi.
1543 Pathway Dr, Naperville

3. Modify the activity to respond to menu selections.

a. Add the code for responding to the menu selections from the user.

Open the file “src/com.garagze/MainActivity.java” and override the “onOptionsItemSelected” method.

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
    case R.id.mi_add_event:
        addEvent();
        return true;
    case R.id.mi_show_map:
        showMap();
        return true;
    case R.id.mi_prefs:
        showPrefs();
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
```

4. Add code for implementing each of the menu selections

a. Add the following methods to MainActivity.java:

```
private void showPrefs() {
      Log.v("MainActivity", "Running showPrefs method.");
}

private void showMap() {
```

```
            Log.v("MainActivity ", "Running showMap method.");
      }

      private void addEvent() {
            Log.v("MainActivity ", "Running addEvent method.");
      }
```

Note: Run the app again and select each of the menu options. The application will write a log message when a menu item is clicked by the user. In subsequent labs we'll be adding the functionality for each menu option.