Chapter 5: Investigate!
Lists, Arrays, and Web Browsers
Objectives

In this chapter, you learn to:

• Create an Android project using a list
• Develop a user interface that uses ListView
• Extend the ListActivity class
• Use an array to create a list
• Code a setListAdapter to display an array
• Design a custom ListView layout with XML code
• Display an image with the ListView control
• Change the default title bar text
Objectives (continued)

• Code a custom setListAdapter for a custom layout
• Call the onListItemItemClick method when a list item is selected
• Write code using the Switch decision structure
• Call an intent to work with an outside app
• Open an Android Web browser
• Launch a Web site through the use of a URI using an Android browser
• Test an application with multiple decisions
Creating a List

• Lists are one of the most common designs in mobile apps
  – Scrollable
  – Selectable
  – Programmable to bring up the next Activity (screen)

Figure 5-1 The San Francisco City Guide Android app
Creating a List

Figure 5-2 Alcatraz and Ferry Marketplace Web Sites

Figure 5-3 San Francisco attractions
Creating a List (continued)

• Steps to complete the app:
  1. Create a list using a ListView control.
  2. Define an array to establish the items of the list.
  3. Add the images used in the project.
  4. Define an XML file to design the custom list with a leading image.
  5. Code a switch decision structure to handle the selection of items.
  6. Open an Android Web browser to display a specified Uniform Resource Identifier (URI).
  7. Create multiple classes and XML layout files to display pictures of attractions.
Creating a List (continued)

– Opening screen contains a vertical list of attractions
– List is automatically scrollable if it exceeds the window size
– ListView is better than TableLayout View because each row can be selected for further action
– Selecting an item opens up a related Web page or an image of the attraction
Creating a List (continued)

- Extending a ListActivity
  - A ListActivity class is needed to display a list of items
  - An ExpandedListView can be used to provide a two-level list

Figure 5-4 ListView control on the Palette

Figure 5-6 Main extends ListActivity

Figure 5-6 Main extends ListActivity
Creating a List (continued)

• Creating an Array
  – **Array variables** can store more than one value
  – Different from other data types that can hold only one value
  – Each individual item in an array is called an **element**
  – Refer to each element using an index in the array

<table>
<thead>
<tr>
<th>Element</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attraction[0]</td>
<td>Alcatraz Island</td>
</tr>
<tr>
<td>Attraction[1]</td>
<td>Ferry Marketplace</td>
</tr>
<tr>
<td>Attraction[2]</td>
<td>Golden Gate Bridge</td>
</tr>
<tr>
<td>Attraction[3]</td>
<td>Cable Car Trolley</td>
</tr>
<tr>
<td>Attraction[4]</td>
<td>Fisherman’s Wharf</td>
</tr>
</tbody>
</table>

Table 5-1 Attraction array with index values
Creating a List  (continued)

• **Declaring an Array**
  – Square brackets [ ] are used to define an array
  – Curley braces {} contain the list of items in the array

```java
String[][] attraction=
{"Alcatraz Island",
"Ferry Marketplace",
"Golden Gate Bridge",
"Cable Car Trolley",
"Fisherman's Wharf"};
```
Creating a List  (continued)

- **Using a setListAdapter and Array Adapter**
  - An **adapter** provides a model for the layout and converts the data into a list
  - The **setListAdapter** connects the list items to the images or Web pages they represent
  - An **array adapter** supplies the array data to the ListView

```java
setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, attraction));
```
Creating a List (continued)

• Adding the Images to the Resources Folder
  – Images must be located in the drawable-hdpi folder
  – Remember that images may be subject to copyright laws

Figure 5-10 Images copied
Creating a List (continued)

- Creating a Custom XML Layout for ListView

![Code Snippet](image)

Figure 5-12 ImageView XML code
Creating a List (continued)

• Creating a Custom XML Layout for ListView

Figure 5-13 TextView XML code
Creating a List (continued)

- Changing the Title Bar Text
  - Custom titles can appear in the title bar at the top of the window

Figure 5-14 Title bar text is changed from default
Creating a List (continued)

• Coding a setListAdapter with a Custom XML Layout
  – Built-in layout is called simple_list_item_1
Creating a List (continued)

• Using the `onListItemClick` method
  – `onListItemClick()` is called when an item from the list is selected
  – The item’s position in the list is captured so the app knows which of the items was selected
  – The position represents the number of the item in the list
Creating a List  (continued)

- **Decision Structure – Switch Statement**
  - If statements are also decision structures
  - The `switch` statement is used when there are many list items to be evaluated
  - Can only evaluate integers or single characters
  - The keyword `case` is used to test each item
  - The keyword `break` is used to exit the switch decision structure
Creating a List (continued)

- **Android Intents**
  - Android intents send and receive activities and services including:
    - Opening a Web page
    - Calling a phone number
    - Locating a GPS position on a map
    - Posting notes to a note-taking program
    - Sending a photo
    - Posting to a social network
Creating a List (continued)

• Launching the Browser from an Android Device
  – The intent sends the browser a URI (Uniform Resource Identifier)
  – URI is similar to URL (Uniform Resource Locator)
  – URI has additional information necessary for gaining access to the resources required for posting the page
  – The action called ACTION_VIEW (must be in CAPS) is what actually displays the page in the browser
  – ACTION_VIEW is the most common action performed on data
Creating a List  

(continued)

• Designing XML Layout Files
  – XML layout files must be designed to display an ImageView control with an image source file

Figure 5-21 Code for launching the ferry Marketplace Web site

```java
23  case 0:
24      startActivity(new Intent(Intent.ACTION_VIEW,
25                  Uri.parse("http://alcatrazcruises.com/")));  
26      break;
27  case 1:
28      startActivity(new Intent(Intent.ACTION_VIEW,
29                  Uri.parse("http://ferrybuildingmarketplace.com")));  
30      break;
```

Opens Web browser to display Ferry Marketplace site
Creating a List (continued)

Figure 5-23 bridge.xml layout file

Figure 5-24 trolley.xml layout file

Figure 5-25 wharf.xml layout file
Adding Multiple Class Files

- A new class is needed to display each image
- Classes must be referenced in the Android Manifest File

Code for Wharf.java class (Bridge and trolley already created)

Figure 5-28 Complete code for Wharf.java class
Creating a List (continued)

- Opening the Class Files
  - startActivity opens the next Activity which launches the appropriate X

- Running and testing the Application
  - Be sure to test every option
  - Should handle all clicks

Figure 5-30 Complete code for Main.java
Summary

• The Java View class creates a list and makes it scrollable; use a ListView control to select each row for further action
• Extend the ListActivity class in Main.java to display a ListView control
• Declare list items in an array variable
• The index provides access to each element in the list (the list begins with 0)
• To declare an array, specify the data type followed by the values
• Use an adapter to display the values in the array
• A ListView control is a container for the list items and the adapter binds the elements of the array to the ListView layout statements if the condition is false
• Drag controls from the palette to the emulator for a simple design
• Add code to the main.xml file to get a custom layout
Summary (continued)

• App names are displayed in the app title bar – can be customized
• setListAdapter has 3 parameters:
  – this class
  – The layout used to display the list
  – The array containing the list values
• Code the onListItemItemClick method to respond to the event of the user’s selection
• Use the switch decision structure with a list or menu
Summary (continued)

- Android intents send and receive activities and services
- Test every possible combination of clicks, including incorrect user entries before publishing the app