

# MCA253 - Mobile Applications

## Unit:3.1

## Displaying Pictures and Menus with Views

**Dr. Siddesha S** MCA, M.Sc Tech (by Research) , Ph.D.  
Asst. Professor,  
Dept. of Computer Applications,  
JSS Science and Technology University  
Mysuru – 570 006

# Using Image Views to Display Pictures

- So far, all the views seen till now are used to display text information.
- However, you can use the ImageView, ImageSwitcher, and GridView views for displaying images.

## ImageView View

- The ImageView is a view that shows images on the device screen.
- To add an image – drag and drop the images into the **res/mipmap** folder.



butterfly.png

ic\_launcher.png

Program: **ImageView**

# ImageSwitcher

- To apply some animation to an image when it transitions from one image to another.
- In this case, the **ImageSwitcher** is used.

## Program: **ImageSwitcher**

- To use the ImageSwitcher view, the **ViewFactory** interface need to be implemented, which creates the views for use with the ImageSwitcher view.
- For this, implement the **makeView()** method need to be implemented
- This method creates a new View to be added in the ImageSwitcher view, which in this case is an **ImageView**.
- In the **onCreate()** method, you get a reference to the **ImageSwitcher view** and set the **animation**, specifying how images should “**fade**” in and **out** of the view.
- Finally, when an image is selected from the Gallery view, the image is displayed in the ImageSwitcher view.

# GridView

- The GridView shows items in a two-dimensional scrolling grid.
- We can be use the GridView together with an **ImageView** to display a series of images.

**Program: MyGridView**

# Using Menus with Views

- **Menus** are useful for displaying additional options that are not directly visible on the main user interface (UI) of an application.
- There are two main types of menus in Android:
  - **Options menu**—This menu displays information related to the current activity. In Android, you activate the options menu by pressing the Menu button.
  - **Context menu**—This menu displays information related to a particular view on an activity. In Android, you tap and hold a context menu to activate it.

# Creating the Helper Methods

- Before you go ahead and create your **options** and **context** menus, you need to create **two helper methods**.
- One **creates a list of items to show inside a menu**, whereas the other **handles the event that is fired when the user selects an item inside the menu**.

## Program: **MyMenus**

- The preceding example creates two methods:
  - `createMenu()`
  - `menuChoice()`
- The `createMenu()` method adds a series of menu items to a `Menu` argument.

# Creating the Helper Methods

- To add a menu item to the menu, you create an instance of the MenuItem class and use the Menu object's `add()` method:

```
MenuItem mnul = menu.add(0, 0, 0, "Item 1");  
{  
mnul.setAlphabeticShortcut('a');  
mnul.setIcon(R.mipmap.ic_launcher);  
}
```

- The four arguments of the `add()` method are
  - `groupId`—The group identifier of which the menu item should be a part. Use 0 if an item is not in a group.
  - `itemId`—A unique item ID.
  - `order`—The order in which the item should be displayed.
  - `title`—The text to display for the menu item.

# Context Menu

- In addition to the options menu, you can also display a context menu.
- A context menu is usually associated with a view on an activity.
- A context menu is displayed when the user taps and holds an item.  
For example, if the user taps a Button view and holds it for a few seconds, a context menu can be displayed.
- If you want to associate a context menu with a view on an activity, you need to call the `setOnCreateContextMenuListener()` method of that particular view.



# Context Menu

- In the preceding example, you call the `setOnCreateContextMenuListener()` method of the Button view to associate it with a context menu.
- When the user taps and holds the Button view, the `onCreateContextMenu()` method is called.
- In this method, you call the `createMenu()` method to display the context menu.
- Similarly, when an item inside the context menu is selected, the `onContextItemSelected()` method is called, where you call the `menuChoice()` method to display a message to the user.
- Notice that the shortcut keys for the menu items do not work. To enable the shortcuts keys, you need to call the `setQueryMode()` method of the Menu object

# Using WebView

- Android SDK provides some additional views that make your applications much more interesting.

## WebView:

- The WebView enables you to embed a web browser in your activity.
- This is very useful if your application needs to embed some web content, such as maps from some other providers, and so on.

## Program: **WebView**

- To use the WebView to load a web page, you use the **loadUrl()** method and pass a URL to it.
- To display the built-in zoom controls, you need to first get the **WebSettings** property from the WebView and then call its **setBuiltInZoomControls()** method.