**Final Project Planning**

**Final Planning Discussion Due 5/1, Final App & Document Due 5/15**

**Final Project Instructions:**

For your final project you will submit two items:

1. Using what you have learned, develop and submit an app. You must demonstrate at least one technique from each chapter plus one more of your choice. You may make the app as simple or as complicated as you wish to demonstrate your skill. Be creative!
2. You will submit a Word document which details how you created the app, the techniques and methods used (in a list), what you found easy and what you found challenging. At the beginning of the document describe the app as if you were listing it on Google Play along with the price and any other relevant information.
3. As you begin planning this project you will post your ideas and progress on the Final Project Planning Discussion. You will also comment and help your classmates as possible. That discussion is worth 40 points.

The Discussion is worth 40 points.

Each part (the app and the document) is worth  200 points for a total of 400 points.

The link to submit the final is in Week 15, Good Luck, I expect to be dazzled!

|  |
| --- |
| **Ch 1: Voila! Meet the Android** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Specify the use of layout and widget controls in the user interface. |
|  |
| **Wrap It Up – Chapter Summary** |
| * To design a user interface, create a layout, which is a container that displays widgets such as TextView, Button, and Checkbox controls, also called objects. |
| * The text property can be updated using the Translations Editor opened from the strings.xml file in the values folder. |
|  |
| **Case Project 1-1: Famous Technology Quotes App** |
| The opening screen displays the famous technology quote of the day. |

|  |
| --- |
| **Chapter 2: Simplify! The Android User Interface** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Develop a user interface using the TextView, ImageView, and Button controls |
| * Add text in strings.xml using the Translations Editor |
| * Create an Android project that includes a Button event. |
| * Create multiple Android Activities |
| * Add a Java class file |
| * Write code using the onCreate method |
| * Display content using the setContentView command |
| * Open a second screen using a Button event handler |
| * Use an OnClickListener to detect interaction |
| * Launch a second screen using a startActivity method |
|  |
| **Case Project 2-1** |
| In a youth hostel reservations app, a hostel is selected and an address and other information is displayed. |
| 1. The opening screen displays the name of a hostel, an image, and a Button control. Research a real name of a hostel and address and cost range to display in your own customized app. |
| 1. When the user selects a hostel, an address and a cost range are displayed in a second screen. |

|  |  |
| --- | --- |
| **Chapter 3: Engage! Android User Input, Variables, and Operations** | |
|  | |
|  | |
|  | |
| **In this chapter, you will learn to…** | |
| * Use an Android theme | |
| * Add text to the String table | |
| * Add an XML array string to strings.xml | |
| * Develop a user interface using Text Fields | |
| * Display a hint using the hint property | |
| * Develop a user interface using a Spinner control | |
| * Add a prompt to a Spinner control | |
| * Declare variables to hold data | |
| * Code the GetText() method | |
| * Understand arithmetic operations | |
| * Convert numeric data | |
| * Code the SetText() method | |
|  | |
| **Case Project 3-1: Catalina Island Boat Express App** |
| Catalina Express has 30 daily departures between Long Beach and Catalina Island. Create a simple app that determines how many boat tickets the user needs and whether the ticket is for going to Catalina Island or heading back to Long Beach. The app displays the total prince for the fare in one direction. | |
| 1. The app displays a title; an image; and a Text Field, Spinner, and Button control. The two options in the Spinner control include To Catalina Island and To Long Beach. Each single passenger ticket is $34 for one way. | |
| 1. When the user taps or clicks the Button control, the number of tickets and the total cost of the fare is displayed. | |

|  |
| --- |
| **Chapter 4: Explore! Icons and Decision-Making Controls** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Create an Android project with a custom icon |
| * Change the text color in controls using hexadecimal colors |
| * Align controls using the gravity properties |
| * Determine layout with the layout:margin properties |
| * Place a RadioGroup and RadioButtons in Android applications |
| * Write code for a RadioGroup control |
| * Make decisions using an If statement |
| * Make decisions using an If Else statement |
| * Display an Android toast notification |
| * Test the isChecked property |
| * Make decisions using nested If statements |
|  |
| **Case Project 4-1: Phone Photo Prints App** |
| The app determines the cost of printing photos from your phone. The pictures are delivered directly to your home. |
| 1. The opening screen requests the number of photos to print from user’s phone. |
| 1. The user selects a radio button labeled 4x6 (19 cents each) or 5x7 (49 cents each) and 8x10 (79 cents each) then selects the ORDER PRINTS button. |
| 1. The cost is displayed for the number of prints. |
| * *The result is rounded to the nearest penny.* |
| * *Do not enter more than 50 prints.* |
| * *Use a theme that displays an Action bar with the custom Action bar icon in the finished layout.* |

|  |
| --- |
| **Chapter 5: Investigate! Android Lists, Arrays, and Web Browsers.** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Create an Android project using a list |
| * Develop a user interface that uses a 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 |
| * Code a custom setListAdapter for a custom layout |
| * Call the onListItemClick 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 website with a URI using an Android browser |
|  |
| **Case Project 5-1: Beach and Mountain Bike Rental App** |
| A bike rental shop would like an app that displays information about their beach and mountain bike rental services. As each bike is selected, a rental bike is displayed. |
| 1. The opening screen displays a list of bikes for rent: beach bikes, mountain bikes, and full bike rental shop website. |
| 1. When the user selected an item from the list, a full-screen image of the item is displayed for the first two bike rentals. The their option opens the website. |
| * *Use the built-in layout simple\_list\_item\_1* |
| * *Use the switch decision structure.* |
| * *Use a String table for image descriptions.* |

|  |
| --- |
| **Chapter 6: Jam! Implementing Audio in Android Apps** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Create an Android project using a splash screen |
| * Design a TextView control with a background image |
| * Pause the execution of an Activity with a timer |
| * Understand the Activity life cycle |
| * Open an Activity with onCreate() |
| * End an Activity with finish () |
| * Assign a class variables |
| * Create a raw folder for music files |
| * Play music with a MediaPlayer method |
| * Start and resume music playback using the start() and pause() methods |
| * Change the Text property of a control |
| * Change the visibility of a control |
|  |
| **Case Project 6-1: Celtic Songs App** |
| A music app compares the different types of Celtic music. |
| 1. A splash screen opens and displays the celtic.png image with the title “Celtic Sounds” for four seconds |
| 1. Two types of celtic music are available in this app. An Irish jig names jig.mp3 can be played while displaying an image of Irish dancers (jig.png). A second selection of bagpipe music plays bagpipes.mp3 while displaying an image of a man playing bagpipes. |
| * *The music should be played and paused by a button control* |
| * *When a song is playing, the other button should not be displayed.* |

|  |
| --- |
| **Chapter 7: Reveal! Displaying Pictures in a GridView** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Create an Android project using a GridView control |
| * Add a GridView to display a two-dimensional grid of images |
| * Reference images through an array |
| * Create an ImageAdapter class |
| * Code an OnItemClickListener |
| * Display a custom toast message |
| * Define a Context resource |
| * Return a value from a method |
| * Determine the length of an array |
| * Assign an ImageView control using setImageResource |
| * Change the scale and layout size of the GridView |
|  |
| **Case Project 7-1: Quick Healthy Snack Ideas App** |
| The Snack app displays five healthy snack options. |
| 1. The screen displays five snacks with three on each row in a GridView control |
| 1. When the user selects a thumbnail image of a healthy snack, a larger image appears below the GridView. |
| * Display each image in the GridView using a layout height of 400dp, two columns, 2dp for horizontal and vertical spacing, and a column width of 150dp. |

|  |
| --- |
| **Chapter 8: Design! Using a DatePicker on a Tablet** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Create an Android project on a tablet |
| * Add a custom launcher and tablet theme |
| * Understand the Calendar class |
| * Use date, time, and clock controls |
| * Determine the system date |
| * Display a DatePicker control |
| * Launch a dialog box containing a DatePicker control |
| * Code an OnDateSetListener method to await user interaction |
| * Determine the date entered on a calendar control |
|  |
| **Case Project 8-1: Appalachian Trail Festival Tablet App** |
| Displays title and event description. When the user taps a button, a calendar for reserving a ticket for the festival appears. The data is then shown as available for a reservation. |
| 1. The opening tablet screen displays an image, a title, an event description, and a button to create a reservation for a day at the festival. |
| 1. When the user taps a button, a DatePicker is displayed in a dialog box. The dialog box allows the user to select the date to attend the year-long festival. |
| * *Use a table layout with four rows within a Linear layout* |

|  |
| --- |
| **Chapter 9: Customize! Navigating with a Master/Detail Flow Activity** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Understand responsive design for Android apps |
| * Create an Android tablet using an application template |
| * Understand the Master/Detail Flow template |
| * Modify the Master/Detail Flow template |
| * Add a WebView control |
| * Display a Web browser within a tablet app |
| * Add an internet permission to the Android Manifest |
| * Customize the content of the sample template file |
| * Display a custom layout in the details pane |
|  |
| **Case Project 9-1: Oasis Spa Tablet App** |
| Describes the services of a full-service spa names Oasis Spa |
| 1. The opening screen displays three list items titled Spa Services, Spa Address, and Full Web Site. The first list item displays two table rows within a table layout with an image in each row with text. |
| 1. The second item displays the address and phone number of Oasis Spa |
| 1. The third list item opens website in a browser. |
| * *Use the Master/Detail Flow template* |

|  |
| --- |
| **Chapter 10: Move! Creating Animation** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Create an Android application with Frame and Tween animation |
| * Understand Frame animation |
| * Understand Tween animation |
| * Add an animation-list XML file |
| * Code the AnimationDrawable object |
| * Set the background Drawable resource |
| * Launch the start() and stop() methods |
| * Add Tween animation to the application |
| * Create a Tween XML file that rotates an image |
| * Determine the rotation pivot, duration, and repeat count of a Tween animation |
| * Load the startActivity Tween animation in a second Activity |
|  |
| **Chapter 10 Project: Northern Lights Animation** |
| Displays multiple photos of the Iceland Aurora Borealis during a single night |
| 1. The app includes a Start Frame Animation button that reveals the animated images frame-by-frame. |
| 1. When the user taps the Stop Frame Animation button, the frame-by-frame animation stops and the last image of the lights rotates 6 times using Tween animation. |

|  |
| --- |
| **Chapter 11: Discover! Persistent Data** |
|  |
|  |
|  |
| **In this chapter, you will learn to…** |
| * Create an Android project using persistent data |
| * Understand different types of persistent data |
| * Understand SharedPreferences persistent data |
| * Understand internal storage |
| * Understand external storage |
| * Understand saving data using a network connection |
| * Understand saving to a database connection |
| * Write data using a SharedPreference object |
| * Instantiate a SharePreference object |
| * Write data using getString() method |
| * Retrieve data from a SharedPrefence object |
| * Read data using a putString() method |
| * Display an ImageView control using code |
|  |
| **Case Project 11-1: BMI Calculator App** |
| A body mass index calculator app computes your BMI using a formula |
| 1. The first Activity opens displaying the image with the title |
| 1. The first screen requests your weight in pounds to the nearest whole pound and your height in inches to the nearest whole inch. These values are saved in persistent data using SharedPreferences. |
| 1. The second Activity opens and retrieves the saved values. |
| 1. The formula is calculated. The BMI is displayed to one-tenth of a decimal place and the image is displayed. |
| **Chapter 12: Finale! Publishing Your Android App** |
|  |
| **In this chapter, you will learn to…** |
| * Understand Google Play |
| * Target various device configurations and languages |
| * Prepare your app for publishing |
| * Create an APK package by exporting an app |
| * Prepare promotional materials |
| * Publish your app on Google Play |