ALL PRACTICAL LABS

Lab01: Welcome to CS102

In this practical lab, students will develop a simple layout (screen) including an ImageView (display college logo), three TextViews (display strings), and a TextClock showing current time.

Upon the completion of the practical lab, students learn:

How to design Graphic User Interface (GUI) in Android:

- Use Relativelayout
- Change the Background colour: use color, xml background, or image
- Use Visual Controls: TextView (display strings/text), ImageView (display image), EditText, Button
- Set TextView/ImageView/Button properties: text, background, style, size, fontFamily
- Change the App style & theme, and “app_name”
- Set XML “View” properties: padding, margin, gravity, match_parent, wrap_content, ...

How to code java in Android:

- Use Toast class to pop up a message on screen;
- Set Click Listener for Visual controls;
- Use AlertDialog class to add an Alert Dialog;

Best practices to develop an Android app:

- Create *color variables* in colors.xml file
- Create *string variables* in strings.xml file
- Create *background drawable* in /drawable/ folder
- Create *menu's items* in /menu/ folder

Student will create the first app called “Welcome to CS102”:

- Part 1: Create a new Android Project with Android Studio IDE
- Part 2: Layout in Android
- Part 3: Add TextView element
- Part 4: Change app-style and app-theme
- Part 5: Add ImageView element
- Part 6: Add EditText element
- Part 7: Add Button element
- Part 8: Investigate Android Manifest

Lab02: Personality Difference App

In this practical lab, students will learn how to develop a simple app by converting an article (“12 Illustrations Showing How Introverts and Extroverts See the World”) into an Android Mobile App.

Upon the completion of the practical lab, students learn:

How to design Graphic User Interface (GUI) in Android:

- Use LinearLayout and how to design nested layout
- Use Visual controls: Button, TextView, ImageView

How to code java in Android:

- Learn to make the connection between visual elements on Layout and Java code in Activity: Find view by id (visual controls) on Layout: findViewbyID()
- Learn to handle user event: Set Click listener and respond for visual controls
- Learn to use Java array of data: 1D-array
- Learn to use “Resources” in Android : Colors & Strings & Drawable
- Learn to set up “Internet connections” to download image, audio and text files in Android

Student will create the app called “Personality Difference”:

- Part 1: Analyse app function and design its user interface

- Part 2: Implement user interaction: when users click “Next” Button, the app will display the next “illustration” image and its associated “caption”
- Part 3: Download Image from Internet (URL) and load it to ImageView: Network connection & Internet in Android
- Part 4: Implement a new app: “10 Devious Scamming Tricks”
- Part 5: Implement a new app: “How Long Does It Take to Make an App? 5 Phases of Mobile App Development Lifecycle”

Lab03: Personality Test App

In this practical lab, students will learn how to develop a simple app by converting an article (“This Mind-Blowing Picture Test Will Reveal Your True Personality”) into an Android Mobile App.

Upon the completion of the practical lab, students learn:
How to design Graphic User Interface (GUI) in Android:
- Use both RelativeLayout & LinearLayout and learn how to design nested layout
- Use Visual controls: TextView, ImageView, Button, Radio Buttons
- Learn how to separate User Interface Update and Back-end processing Use multiple “Thread” in Android

How to code java in Android:
- Make connection between java variables and visual elements on Layout;
- Set Click listener and respond for visual controls
- Use Resources in Android: colors, strings, dimension;
- Use multiple “Thread” in Android: Handler class, Runnable Interface;
- Use “Dialog” API to pop up a window on screen;

Lab04: Animation College Program Manager App

This practical lab shows students how to work with multiple-activity and How to exchange data between those Activities by using “Intent” object.
In this tutorial, students will develop a mobile application called Animation College (AC) Program Management. All the information of programs and courses offered by AC are available on AC and AMES websites:
AC website: http://www.animationcollege.co.nz/
AMES website: http://www.ames.ac.nz/#welcome-to-ames

Upon the completion of the practical lab, students learn:
How to design Graphic User Interface (GUI) in Android:
- Use both RelativeLayout & LinearLayout and learn how to design nested layout;
- Use Visual controls: TextView, ImageView, Button, Radio Buttons, WebView, ListView, Spinner (drop-down menu);
- Learn how to use ListView and Spinner for displaying an array of items;

How to code java in Android:
- Learn how to integrate and modify the “Activity lifecycle” in the app: onCreate -> onStart -> onResume -> onPause -> onStop -> onDestroy or onRestart;
- Learn how to create Multiple-Activity app: use “Intent” object to navigate amongst Activities;
- Learn How to use ListView element and Adapter to populate items inside ListView;
- Learn to use XML Parser technique to process data stored in XML file;
- Learn to use SMS API in Android for sending SMS
- Learn to use “implicit Intent” to send email in Android

Student will create the app called “ACProgramManager”:

- Part 1: Analyse app function and design its main user interface
- Part 2: Add the second Activity: “AboutAMES_Activity”
- Part 3: Add the third Activity: “AboutAC_Activity”
- Part 4: Add the fourth Activity: “ListViewProgrammes_Activity”
- Part 5: Add the fifth Activity: “ProgrammeDetail_Activity”
- Part 6: Add the sixth Activity: “Enrol_Activity”
- Part 7: Add the seventh Activity: “ContactUs_Activity”

Lab05: Group Chat App Using Firebase

Mobile applications need a back-end server in order to perform tasks such as authenticating users and synchronizing user data across multiple devices. Creating such a server, however, requires a skill set that most independent app developers lack. Fortunately, there are several back-end as a service platforms, often called BaaS, you can use today. Google's Firebase is one such platform. Firebase offers essential services such as analytics, crash reporting, user authentication, and cloud messaging at no cost. Its freemium services include a real-time NoSQL database, file hosting, and static website hosting.

In this tutorial, students learn how to use Firebase UI to create a group chat app (a real-time social application) you can share with your friends. It's going to be a very simple app with just one chat room, which is open to all users.
As you might have guessed, the app will depend on Firebase Auth to manage user registration and sign in. It will also use Firebase's real-time database to store the group chat messages. You also learn how to add user authentication, analytics, and remote real-time cloud data storage to your Android app using Firebase.

Upon the completion of the practical lab, students learn:
How to design Graphic User Interface (GUI) in Android:

- Use both RelativeLayout & LinearLayout and learn how to design nested layout;
- Use Visual controls: TextView, FloatingActionButton, Button, TextInputLayout, EdiitText, ListView;
- Learn how to use ListView and Spinner for displaying an array of items;

How to code java in Android:

- Learn to integrate the third party library in the app;
- Learn to create Firebase project & how to integrate Firebase project to Android project
- Learn how Use Firebase Auth to handle user login and registration processes
- Learn how to use read and write to real-time Database
- Learn How to use ListView element and Adapter to populate chat “messages” inside ListView;

Student will create the app called “Group Chat App”:

- Part 1: Create Android Studio Project
- Part 2: Create Firebase database and integrate it to Android Project
- Part 3: Use Firebase Auth to handle user login and registration processes
- Part 4: Use Real-time Database to handle group chat

Lab06: Animation College Program Manager App using Cover Flow

Students have developed a “typical” and “standard” Animation College Programme Management (ACPM) App in previous tutorial. In this practical lab, students learn to make ACPM app more fancy and professional by using CoverFlow library. Cover-flow Layout is an exciting topic in interface designing. It can act like a ViewPager with swiping to change content view, but also, the center item (selected one) is larger than another. This view style gives a 3D effect, easy to manipulate with the elements.

In this tutorial, students implement a powerful library to make a flow (carousel) layout for “View all Programmes” by using third-party library FeatureCoverFlow.

Upon the completion of the practical lab, students learn:

How to design Graphic User Interface (GUI) in Android:

- Use both RelativeLayout & LinearLayout and learn how to design nested layout;
- Use Visual controls: FeatureCoverFlow, TextView, TextSwitch, ImageView, ListView, Spinner (drop-down menu);
- Learn how to use ListView and Spinner for displaying an array of items;
- Use “Dialog” API to pop up a window on screen;

How to code java in Android:

- Learn to integrate the third party library in the app;
- Learn to use FeatureCoverFlow and populate into it;
- Learn How to use ListView element and Adapter to populate items inside ListView;
- Learn to use XML Parser technique to process data stored in XML file;

Student will create the app called “ACProgramManagerUsingCoverFlow”:

- Part 1: Analyse app function and design its main user interface: “carousel layout” & Add dependencies to Gradle
- Part 2: Add “Programme” class: Design “programme_view” layout
- Part 3: Add layout for customizing Dialog layout
- Part 4: Add a customized Adapter for FeatureCoverFlow element
- Part 5: Programming MainActivity
- Part 6: Upgrade “Programme Detail Dialog” to show all detailed information
- ….