Name of the Programme: MCA
Course code: CSA-606
Title of course: Mobile App Development
Number of Credits: 4 (2L-2T-0P)
Effective from AY: 2022-23

Prerequisites for the course	Knowledge of OS and networking; and web development basics	
Objectives	On completion of this course, the learner should be able to successfully build, debug and deploy android apps.	
Content	**Android OS, Ecosystem & Basics** ● Mobile Platforms & OSs; Approaches to mobile development; Android OS; Android System Architecture; Android App Lifecycle; Play Store ● Intro; Create Your First Android App; Layouts, Views and Resources; Text and Scrolling Views; Resources to Help You Learn ● Debugging your apps; Testing your app; Support libraries, and Backwards Compatibility.	6 hours
	User Interface & Lifecycle ● Screen Sizes; User Interaction - User Input Controls, Menus; Screen Navigation; RecyclerView ● Delightful User Experience; Drawables, Themes and Styles; Material Design; Providing Resources for adaptive layouts ● Testing the User Interface ● Activities and Intents; The Activity Lifecycle and Managing State; Starting Activities with Implicit Intents	14 hours
	Background Tasks & Notifications ● Background Tasks; AsyncTask and AsyncTaskLoader; Connecting to the Internet; Broadcast Receivers; Services ● Triggering, Scheduling, and Optimizing Background Tasks; Notifications; Alarm Manager; Transferring Data Efficiently.	4 hours
	Data Saving, Retrieving, Loading ● Overview to storing data ● Shared Preferences; App Settings ● SQLite; Firebase ● Sharing Data: Content Resolvers and Content Providers ● Using Loaders to Load and Display Data ● Connecting with API service endpoints.	6 hours
	Suggested Sample List of Assignments:- 1) Build an OO system (like elevators in a building, EVM, etc.). Employ use of design patterns (like Adapter, Singleton, Observer, etc.) 2) Creating a Java/Kotlin project using build tool (e.g. Gradle, Maven) 3) Create a hello world android app using IDE (preferably Android Studio). Try deploying on emulator/mobile. Debug using logcat. 4) Create a calculator app (similar to the app installed in the device used during development) 5) Using intents create a game (like a maze). Explore having raster images & vector graphics in the app. 6) Create a CRUD app. Explore the use of various form elements/widgets and fragments. 7) Create a To-Do app. Explore adding the views/view-groups programmatically (e.g. using inflate, recycler view). Use material design in the UI. 8) Create an app accessing data exposed by another app/ service.	20 hours

	Explore BroadcastReceiver, services, etc. 9) Create an app that will run in background and communicate information through status bar/ push-notifications. 10) Create a CRUD app using data stored locally. Explore ROOM, SQLite 11) Create an app to consume an API and populate the layout with appropriate views. 12) Create an app to contain a webapp.	
	Mini-project Ideally done in a group. It should include design and implementation of an android application. Project implementation should mandatorily use at least 2 mobile-specific functionality (to justify as a mobile app and not web app). The GUI of the app should follow design guidelines (e.g. Material/ Flat Design). Conduct and progress of the project could follow industry practices (e.g. UX mocks, git, scrum, etc.).	10 hours
Pedagogy	Assignments / tutorials / peer-learning / troubleshooting/ case studies	
References/ Readings	● Bill Philips & Brian Hardy, “Android Programming: The Big Nerd Ranch Guide” ● Dawn Griffiths & David Griffiths, “Head First Android Development” ● Ian F. Darwin, “Android Cookbook” ● https://developer.android.com ● https://kotlinlang.org ● https://material.io	
Course Outcomes	1. Learner will understand the android ecosystem, android versions & compatibility across them. 2. Learner will be able to design user interfaces specifically to be run native android devices. 3. Learner will be able to evaluate which type of views & widgets are preferable for various use cases. 4. Learner will be able to build and design navigation flows in an app. 5. Learner will be able to connect the app to Android services or apps already available on the device. 6. Learner will be able to build apps that can store data locally or remotely.	

(Back to top)