Notes – 21 / 09 / 2020 – Klapesh training

Activities and Intents – Codelab excercises and fundamental concepts

* [Application Fundamentals](http://developer.android.com/guide/components/fundamentals.html)
* [Activities](http://developer.android.com/guide/components/activities.html)
* [Understand the Activity Lifecycle](https://developer.android.com/guide/components/activities/activity-lifecycle)
* [Intents and Intent Filters](http://developer.android.com/guide/components/intents-filters.html)
* [Handle configuration changes](http://developer.android.com/guide/topics/resources/runtime-changes.html)
* [Activity](http://developer.android.com/reference/android/app/Activity.html)
* [Intent](http://developer.android.com/reference/android/content/Intent.html)

Viewgroups

Alignment is essential

1 LinearLayout

2 RelativeLayout

3 ConstraintLayout

4 Coordinatorlayout

1. Dataset => Adapter -> UI (Recycleview, listview or spinner)
2. Kotlin in week 3
3. Android arsenal for customer spinners and buttons, etc
4. Better pickers

Room database

Entities

Database

DAO

* Separate initialization into functions - modular

Threading – executable threading tasks.

Database queries must be executed on a separate thread.

RunOnUIthread

Runnable

Run()

Week 2) 28/09/2020

Recycler View

Zepplin

Stencil

Material design

Fragments

Activities the mother of fragments

1 activity and only fragments

Fragments for tabs or user information or scroll tab in an application.

See slides

Place in enclosing layout or viewgroup

Add dynamically

2 friends getSupportFragmentManager

Add a fragment to the container

Commit

Special keys for creating keywords in android studio

Try implementing as many as these as possible

https://developer.android.com/studio/intro/keyboard-shortcuts

OnCreateview() for initialisation

OnViewCreated() initialize UI components

Fragment to Fragment communication via the. Main activity via an interface

Navgraph is a nice navigation tool for visualisation of fragments in your application. Requires specification when a new fragment is created.

Networking

API Parsing

Kotlin - introduction

Higher-order functions

Special key words for classes

Extension functions

Open keyword

Final keyword

Access modifiers like private, protected, public and internal

Elvis operator – Null safety operator

Type declarations after the function and default parameters

Val and var

Kotlin OOP

Classes, constructors, getters and setters

Inheritance

Visibility modifiers

Abstract class

Interfaces

Nested and inner classes

Data class

Sealed class

SOLID principles – extensibility, single responsibility principle, open closed principle, liskov substitution principle, interface segregation principle, dependency injection

Constant keyword

Design patterns in Kotlin

**Billion Dollar Exception** – null pointer exception (NPE) – Kotlin can’t understand null

* Safe Calls
* Elvis operator
* Bang exception
* Let operator

Inline function reduces memory overheads on higher order functions

Kotlin coroutines

Jetpack architecture

Network Bound Resource

Dependency injection – injecting a dependency from one class to another class.

Package examples include Dagger and Coin

Service Locator

Repository pattern

Week 4