|  |  |  |
| --- | --- | --- |
| **Mod-4: Activity(21-30)** | **Start: 9:15 am** | **Stop: 10:45 am** |
| * Content providers can help an application manage access to data stored by itself, stored by other apps, and provide a way to share data with other apps. * They encapsulate the data, and provide mechanisms for defining data security. * Content providers are the standard interface that connects data in one process with code running in another process. * Implementing a content provider has many advantages. * Most importantly you can configure a content provider to allow other applications to securely access and modify your app data. * AsyncTask enables proper and easy use of the UI thread. This class allows you to perform background operations and publish results on the UI thread without having to manipulate threads and/or handlers. * AsyncTask is designed to be a helper class around Thread and Handler and does not constitute a generic threading framework. | | |

|  |  |  |
| --- | --- | --- |
| **Activity(31-42)** | **Start: 10:45 am** | **Stop: 12:30 pm** |
| * A loader that queries the ContentResolver and returns a Cursor. This class implements the Loader protocol in a standard way for querying cursors, building on AsyncTaskLoader to perform the cursor query on a background thread so that it does not block the application's UI. * The Room persistence library provides an abstraction layer over SQLite to allow for more robust database access while harnessing the full power of SQLite. | | |

|  |  |  |
| --- | --- | --- |
| **Activity(43-52)** | **Start: 2:30 pm** | **Stop: 4:35 pm** |
| * The value of the annotation includes the query that will be run when this method is called. This query is verified at compile time by Room to ensure that it compiles fine against the database. * The arguments of the method will be bound to the bind arguments in the SQL statement. * Room only supports named bind parameter :name to avoid any confusion between the method parameters and the query bind parameters. * Room will automatically bind the parameters of the method into the bind argum * There are 3 types of queries supported in Query methods: SELECT, UPDATE and DELETE. | | |

|  |  |  |
| --- | --- | --- |
| **Activity(53-58)** | **Start: 4:35 pm** | **Stop: 5:40 pm** |
| * The Android framework manages the life cycles of UI controllers, such as activities and fragments. * The framework may decide to destroy or re-create a UI controller in response to certain user actions or device events that are completely out of your control. * If the system destroys or re-creates a UI controller, any transient UI-related data you store in them is lost. * A task is a collection of activities that users interact with when performing a certain job. * When the user selects a message, a new activity opens to view that message. This new activity is added to the back stack. * If the user presses the Back button, that new activity is finished and popped off the stack. | | |

|  |
| --- |
| **Deliverable Status** |

Module -5:

<https://github.com/manojbandari/Android/tree/master/module-5>