**Assignment-8**

**Swift-5**

SCADE, the cross platform Swift compiler that can be used to develop native apps for both ARM and X86 Android platforms, has been updated to Swift 5.

The new release also adds support for Autolayout on Android, so developers can now use autolayout to create UIs on both iOS and Android. The updated version can also use third party frameworks, including native C libraries.

If you've not previously encountered this implementation of cross platform Swift, this video shows how to build a Hello World app with an earlier release of SCADE for both Android and iOS:

SCADE applications are coded in Swift, and compiled to native machine code using the Apple Swift 5.x compiler on both iOS and Android. The SDK provides cross platform access to the native controls such as text fields, keyboard, map and camera, and enables the use of these controls through one code base. The major functionality is exposed through one unified API, and for platform specific functionality you can cast the control to its native control class and use it without restrictions.

SCADE includes a vector graphics engine with built-in transformations and animations for special effects. It includes native UI controls, and the engine adjusts the UI  automatically to the screen size of the respective device. It also uses Fusion, a framework that can be used to call Android functionality from Swift. You also get access to Eclipse-based IDEs such as Code Wizard and Visual UI designer and a Scade simulator.

The team behind SCADE says it plans to open source part of it, and finance ongoing operations through paid support models.

Fragments in android

**Android Fragment** is the part of activity, it is also known as sub-activity. There can be more than one **fragment** in an activity. **Fragments** represent multiple screen inside one activity. **Android fragment** lifecycle is affected by activity lifecycle because **fragments** are included in activity.

Adapter in android

**Adapter** is a bridge between UI component and data source that helps us to fill data in UI component. It holds the data and send the data to an **Adapter** view then view can takes the data from the **adapter** view and shows the data on different views like as ListView, GridView, Spinner etc.