|  |
| --- |
| **Part 2**  **Xcode**  Submitted To :  Mr. Manhar Kapoor  Submitted By :  **Rupinder Kaur– Student ID – A00110794**  INTRODUCTION TO XCODE IDE   * XCode is an integrated development environment developed to work on Mac operating systems. * It contains a suite of software development tools developed by apple. * XCode facilitates us to develop software for macOS, tvOS, iOS, and watchOS. * The latest stable release of XCode is 11.0, which is available on the Mac App Store for all the users of macOS Mojave. In this section of the tutorial, we will go through various contexts of XCode. We will also go through multiple sections of XCode. * The latest version of XCode i.e., Version 11, can be installed on macOS Mojave from the Apple Mac App store.   A Quick walkover of XCode   * When we create a new XCode Project, the following window is shown which provides the target information of the XCode Project.   Introduction to XCode IDE   * It shows every information of the project that includes Bundle Identifier, App version, Build Version, Signing Information, Deployment Information, Linked Binaries and framework information, and application launch icons. * Above the target information, there is a pane of project information which shows all the information about the project given in the following image. It contains information about the iOS version for which the application is created. It also contains release information.   Introduction to XCode IDE  COMPONENTS OF AN XCODE WINDOW  Introduction to XCode IDE **Standard Editor** Introduction to XCode IDE  It contains the information about the file commented at the top and the initial View Controller class file with the lifecycle method created. At the topmost pane of the Standard editor, the hierarchical information about the project file is shown. It can also be used to open other files in the same editor. However, we can also navigate to other project files using the project navigator also. **Assistant Editor** Introduction to XCode IDE  The Assistant Editor is mainly used to create outlets of the storyboard components (Textfield, Label, etc.) in the corresponding View Controller class file. However, the Assistant editor facilitates us to look at two files in the editor simultaneously. **Project Navigator** The project navigator is shown at the left of the window. It shows the file structure of the project. It is used to navigate through the project. Initially, an XCode project contains the files shown in the following image.  Introduction to XCode IDE **File Inspector** The inspectors are shown in the right of the XCode window, as shown in the below image. The file inspector shows the full information about the corresponding swift file opened in the standard editor. It contains the Name, Type, Location, and the interface builder document-related information.  Introduction to XCode IDE **Quick help inspector** It is used to provide help to the user where it can search the documentation about the syntax. It is shown as the below image.  Introduction to XCode IDE **Identity Inspector** Identity Inspector is mainly used when we need to work with the storyboard. It shows information about storyboard components (View Controllers) and their corresponding Swift class files. To program the storyboard components accordingly, we need to assign class files to them.  Introduction to XCode IDE **Attribute Inspector** An attribute inspector is used to give some attributes to the corresponding UIView in the storyboard like content mode, tags, interaction, background color, font color, font size, etc. The attributes given to the storyboard views using attribute inspector are static and can be changed programmatically at runtime.  Introduction to XCode IDE **Size Inspector** Size inspector provides information about the size constraints given to the view while designing using a storyboard. We can alter the size constraint given to the view using the Size Inspector.  Introduction to XCode IDE **Connections Inspector** It shows the information about the connections of the corresponding storyboard UIView to the swift class file. It contains all the connections of the storyboard to the swift class files.  Introduction to XCode IDE **Media Library** Using the media library, we can insert the desired widget to the storyboard using drag & drop functionality. We can open the media library using **command + shift + L** short key.  Introduction to XCode IDE  Installing Xcode **Steps to install Xcode**  1. Download Xcode 2. Install the command line tool 3. Open the new version 4. Delete files  **Step 1: Download Xcode**  * There are two ways to do this. For the latest version and a theoretically "easy" installation, you can use the App Store. I don't recommend this option.  **Option 1: Download via the App Store for the latest version** Steps:   * Open the App Store on your mac * Sign in * Search for Xcode * Click install or update  **Option 2: Download via the Developer site for a specific version**  1. Head to the "more" section of the [Apple developer website](https://developer.apple.com/download/more/) 2. Sign in with your iTunes account id 3. Type in the version that you'd like, and download the Xcode\_x\_x\_x.xip file. Keep in mind that Xcode 11.4.1 is 8 gigabytes, so this will take awhile depending on your internet connection. 4. Once the file is downloaded, click on .xip to extract it. Your laptop will extract it to the same folder you downloaded it to. This extraction process is automatic. You don't need to do anything more after you click on the .xip file. This step will take a few minutes. 5. [Optional] Once extracted, rename the application to “Xcode11.x.x” if you are using multiple versions. 6. Drag application to the Applications folder 7. [Optional] Set the new Xcode version as the default. Open Terminal and type sudo xcode-select -switch /Applications/Xcodex.x.x.app . Replace x.x.x with the version number. For example: Xcode11.4.1.app. You will need to enter in your computer admin password. I'm pretty sure this will update the default Xcode version for all users on your computer, so best to check with other users first    **Step #2: Install the command line tool (CLT)** If you have multiple users on your computer, you will need to update the CLT for each user.  **Download .dmg** **Step #3: Open Xcode** Open the Applications folder and open the new version of Xcode. If you renamed Xcode, make sure you open the correct application.  Xcode may prompt you to install additional components. Click install. This will take a few minutes.   **Step #4. Delete the files** If you don't need the older versions of Xcode on your computer, you can uninstall them and get some hard drive space back.  You can also delete the .xip file of the version you just downloaded, as well as the CLT.dmg file.  X CODE CAPABILITIES **Source Editor** Write code using a professional editor with advanced code completion, code folding, syntax highlighting, and message bubbles that display warning, errors, and other context-sensitive information inline with your code. **Assistant Editor** The Assistant button splits the editor in two, creating a secondary pane that automatically displays files that are most helpful to you based on the code you are actively editing. It can show the header counterpart, the superclass, callers, callees, or other helpful files. **Interface Builder Built-In** Design and test your user interface without writing a line of code, prototype in minutes, then graphically connect your interface to the source within the Xcode editor. **Integrated Build System** Handles the most complex builds, scaling to maximize the power of multi-core Macs, and will automatically sign, provision, and install iPad and iPhone apps onto a device. **Quick Help** Shortened API documentation is displayed while you’re programming, including comments that you write for your code. A brief overview is presented during code completion, with more links and references available within the Utility area. **Graphical Debugger** Debug your app directly within the Xcode editor. Hover over any variable to drill into its contents, use Quick Look to see the data it contains, or right-click to add the variable to the watch list. **Declarative syntax** Write simpler code with a declarative Swift syntax that clearly states what your user interface should do. **Design tools** Drag and drop to construct or edit your interface. Quickly make changes to visual UI elements with pop-up inspectors. **Data Recording** Tell Instruments which app to analyze, what type of data to collect, and simply click the big red button as data is collected and stored for further analysis. |