Government of Pakistan

**National Vocational and Technical Training Commission**

**Prime Minister Youth Skills Development Program**

 "Skills for All"



**Course Contents / Lesson Plan**

**Course Title: iOS 12 & Xcode 10 - Complete Swift 4.2 & Objective-C Course**

**Duration:** 1 Month (4 Weeks)

|  |  |
| --- | --- |
| **Trainer Name** |  |
| **Author Name** | **Muzammil Hasan (Manager Research KICS UET Lahore)** |
| **Course Title** | **iOS 12 & Xcode 10 - Complete Swift 4.2 & Objective-C Course** |
| Objectives and Expectations  | **Employable skills and hands-on practice in iOS 12 & Xcode 10 - Complete Swift 4.2 & Objective-C Course**This is a special course designed to address unemployment in the youth. The course aims to achieve the above objective through hands on practical training delivery by a team of dedicated professionals having rich market/work experience. This course is therefore not just for developing a theoretical understanding/back ground of the trainees. Contrary to that, it is primarily aimed at equipping the trainees to perform commercially in a market space in independent capacity or as a member of a team.The course therefore is designed to impart not only technical skills but also soft skills (i.e. interpersonal/communication skills; personal grooming of the trainees etc.) as well as entrepreneurial skills (i.e. marketing skills; freelancing etc.). The course also seeks to inculcate work ethics to foster better citizenship in general and improve the image of Pakistani work force in particular. **Main Expectations:** In a comprehensive course focusing on iOS 12 and Xcode 10, the primary objectives revolve around equipping participants with employable skills crucial for success in the dynamic field of mobile app development. The course aims to provide a deep understanding of the Swift 4.2 and Objective-C programming languages, which are fundamental to iOS development. By mastering these languages, participants can build robust and efficient applications, ensuring they are well-prepared to contribute effectively to real-world projects.Hands-on practice is a key component of the course, enabling participants to apply theoretical knowledge in practical scenarios. Through guided exercises and projects, students have the opportunity to work directly with iOS 12 and Xcode 10, gaining valuable experience in developing applications for Apple's ecosystem. This hands-on approach not only enhances technical proficiency but also fosters problem-solving skills and a practical understanding of the development process. Participants can expect to leave the course with a portfolio of projects that showcase their ability to create functional and polished iOS applications.Furthermore, the course recognizes the importance of staying up-to-date with the latest technologies and industry practices. With iOS 12 and Xcode 10 being current at the time of the course, participants can expect to learn cutting-edge techniques and best practices in mobile app development. This forward-looking approach ensures that graduates are well-positioned to meet the demands of the ever-evolving iOS development landscape, making them valuable assets in the job market. Overall, the course seeks to bridge the gap between theoretical knowledge and practical application, fostering a well-rounded skill set that enhances employability in the competitive field of iOS development.1. **Motivational Lectures**

The proposed methodology for the training under reference employs motivation as a tool. Hence besides the purely technical content, a trainer is required to include elements of motivation in his/her lecture. To inspire the trainees to utilize the training opportunity to the full and strive towards professional excellence. Motivational lectures may also include general topics such as the importance of moral values and civic role & responsibilities as a Pakistani. A motivational lecture should be delivered with enough zeal to produce a deep impact on the trainees. It may comprise of the following:* Clear Purpose to convey the message to trainees effectively.
* Personal Story to quote as an example to follow.
* Trainees Fit so that the situation is actionable by trainees and not represent a just idealism.
* Ending Points to persuade the trainees on changing themselves.

 A good motivational lecture should help drive creativity, curiosity, and spark the desire needed for trainees to want to learn more.The impact of a successful motivational strategy is amongst others commonly visible in increased class participation ratios. It increases the trainees’ willingness to be engaged on the practical tasks for a longer time without boredom and loss of interest because they can see in their mind's eye where their hard work would take them in short (1-3 years); medium (3 -10 years) and long term (more than 10 years). As this tool is expected that the training providers would make arrangements for regular well planned motivational lectures as part of a coordinated strategy interspersed throughout the training period as suggested in the weekly lesson plans in this document.Course-related motivational lectures online link is available in **Annexure-II**.1. **Success Stories**

Another effective way of motivating the trainees is using Success Stories. Its inclusion in the weekly lesson plan at regular intervals has been recommended till the end of the training. A success story may be disseminated orally, through a presentation, or using a video/documentary of someone that has risen to fortune, acclaim, or brilliant achievement. A success story shows how a person achieved his goal through hard work, dedication, and devotion. An inspiring success story contains compelling and significant facts articulated clearly and easily comprehendible words. Moreover, it is helpful if it is assumed that the reader/listener knows nothing of what is being revealed. The optimum impact is created when the story is revealed in the form of:-* Directly in person (At least 2-3 cases must be arranged by the training institute)
* Through an audio/ videotaped message (2-3 high-quality videos must be arranged by the training institute)

It is expected that the training provider would collect relevant high-quality success stories for inclusion in the training as suggested in the weekly lesson plan given in this document. The suggestive structure and sequence of a sample success story and its various shapes can be seen in **Annexure III**. 1. **Case Studies**

Where a situation allows, case studies can also be presented to the trainees to widen their understanding of the real-life specific problem/situation and to explore the solutions.In simple terms, the case study method of teaching uses a real-life case example/a typical case to demonstrate a phenomenon in action and explain theoretical as well as practical aspects of the knowledge related to the same. It is an effective way to help the trainees comprehend in depth both the theoretical and practical aspects of the complex phenomenon in depth with ease. Case teaching can also stimulate the trainees to participate in discussions and thereby boost their confidence. It also makes the classroom atmosphere interesting thus maintaining the trainee interest in training till the end of the course.Depending on suitability to the trade, the weekly lesson plan in this document may suggest case studies be presented to the trainees. The trainer may adopt a PowerPoint presentation or video format for such case studies whichever is deemed suitable but only those cases must be selected that are relevant and of a learning value. The Trainees should be required and supervised to carefully analyze the cases. For this purpose, they must be encouraged to inquire and collect specific information/data, actively participate in the discussions, and intended solutions to the problem/situation. Case studies can be implemented in the following ways: - 1. A good quality trade-specific documentary ( At least 2-3 documentaries must be arranged by the training institute)
2. Health &Safety case studies (2 cases regarding safety and industrial accidents must be arranged by the training institute)
3. Field visits( At least one visit to a trade-specific major industry/ site must be arranged by the training institute)
 |
| Entry-level of trainees | The iOS 12 & Xcode 10 - Complete Swift 4.2 & Objective-C Course is designed to cater to entry-level trainees who are passionate about delving into the world of mobile app development. Ideal candidates include recent graduates with a degree in computer science or a related field, self-taught programmers looking to formalize their skills, and individuals seeking a career change into the dynamic realm of iOS development.Basic programming knowledge is recommended but not mandatory, making this course accessible to those with limited coding experience. Trainees should have a foundational understanding of concepts like variables, loops, and conditional statements. Additionally, a keen interest in technology, creativity, and problem-solving will enhance the learning experience.* The course is structured to provide a supportive environment for beginners, offering step-by-step guidance through the intricacies of Swift 4.2 and Objective-C. Trainees will benefit from hands-on practice and real-world application, gradually building their proficiency in iOS development. By the end of the course, participants can expect to have a solid foundation in programming for iOS, empowering them to pursue entry-level positions in the field or further advance their skills in more advanced courses.
 |
| **Learning Outcomes of the course** | Upon completion of the IoS Objective-C course, participants can expect to achieve the following learning outcomes:Study guide is taken from Exam PL-300: IoS Objective-C | Microsoft Learn* Full understanding of Xcode 10 and all it has to offer
* Learn how to program for IOS 12 to create your own apps
* Create fully featured games
* Be able to support all devices and screen sizes
* Full explanation of all objects and functions within IOS 12 development
* Develop in Objective-C and Swift 4.2
* Earn revenue from your apps
* Start a new career

The content of this lesson plan is adopted from," ensuring alignment with global standards and practices. For further reference, the link to the source material is provided below: |
| **­­­Course Execution Plan** | The total duration of the course: **1 month (4 Weeks)**Class hours: **4 hours per day**Theory: **20%**Practical: **80%**Weekly hours: **20 hours per week**Total contact hours: **80 hours** |
| **Companies offering jobs in the respective trade** | Several tech companies actively seek professionals with skills in iOS development using Swift and Objective-C. As of my last knowledge update in January 2022, some notable companies that often hire individuals with expertise in iOS 12, Xcode 10, Swift 4.2, and Objective-C include:**Apple** **Inc**.: The creator of the iOS platform, Apple itself is a significant employer of iOS developers. Working directly with Apple technologies and contributing to the development of apps for their ecosystem is a prestigious opportunity.**Google**: Although primarily known for Android, Google also develops and maintains iOS applications. Google's iOS development teams often look for skilled individuals to work on apps like Google Maps, Gmail, and others that are available on the Apple App Store.**Facebook** (Meta): As a major player in the social media and technology industry, Meta (formerly Facebook) frequently hires iOS developers to work on apps like Facebook, Instagram, WhatsApp, and Oculus.**Microsoft**: With a growing presence in the mobile app space, Microsoft hires iOS developers to work on applications like Microsoft Office, Outlook, and Teams, ensuring a seamless experience for iOS users.**Uber**: Ride-sharing and transportation companies like Uber have a constant need for iOS developers to enhance and maintain their mobile applications, ensuring a smooth experience for users.**LinkedIn**: LinkedIn, a subsidiary of Microsoft, often seeks iOS developers to contribute to the development and improvement of their professional networking app.**Airbnb**: As a leading platform in the hospitality industry, Airbnb frequently hires iOS developers to work on their mobile app, providing users with an intuitive and enjoyable experience.**Pinterest**: Social media and visual discovery platform Pinterest hires iOS developers to contribute to the development of their mobile app, allowing users to discover and save ideas. |
| **Job Opportunities** | Job opportunities in iOS development, particularly for those skilled in iOS 12, Xcode 10, Swift 4.2, and Objective-C, are abundant in the tech industry. Here are some common job roles and opportunities you might explore:**iOS Developer**: As an iOS developer, you'll be responsible for designing, developing, and maintaining mobile applications for Apple devices. This role often requires proficiency in Swift, Objective-C, Xcode, and an understanding of iOS frameworks.**Mobile App Developer**: Companies that focus on mobile app development, regardless of platform, may have positions for developers with expertise in iOS. This role could involve working on cross-platform frameworks or native iOS app development.**Software Engineer (iOS):** Many tech companies hire software engineers with a specialization in iOS development. This role may involve collaborating with cross-functional teams to create and optimize software solutions for iOS platforms.**iOS Software Engineer:** Some companies specifically seek iOS software engineers to work on the design and development of iOS applications. This role often involves collaborating with UI/UX designers, product managers, and other team members.**iOS Development Lead:** With experience, you might find opportunities to lead iOS development teams. This role involves guiding and mentoring junior developers, making technical decisions, and ensuring the successful delivery of iOS projects.**Quality Assurance (QA) Engineer (iOS):** QA engineers specializing in iOS testing are crucial to ensuring the quality and functionality of iOS applications. They test applications for bugs, compatibility, and overall user experience.**iOS Architect**: In larger organizations, there might be opportunities for iOS architects who can design and implement the overall structure of complex iOS applications, making high-level design decisions and ensuring best practices.**Freelance iOS Developer:** If you prefer flexibility, freelancing as an iOS developer allows you to take on projects from various clients, working on a contract basis. Platforms like Upwork and Freelancer offer opportunities for freelance iOS developers.Startups and Entrepreneurial Ventures: Joining a startup or entrepreneurialventure provides an opportunity to work on cutting-edge projects, contribute to the development of innovative apps, and potentially take on a variety of roles within a smaller team. |
| **No of Students** | 25 |
| **Learning Place** | Classroom / Lab |
| **Instructional Resources** | **Online Courses:****Udemy - iOS 12 & Swift: The Complete iOS App Development** [**https://www.udemy.com/course/ios-12-xcode-10/**](https://www.udemy.com/course/ios-12-xcode-10/)In-depth course covering Swift, Xcode, iOS development, and app deployment.**Udemy iOS Bootcamp****Stanford University - Developing iOS 11 Apps with Swift**A series of free iOS development courses available on iTunes U.Stanford iTunes U**Coursera - iOS App Development with Swift Specialization**A series of courses from the University of Toronto covering Swift and iOS development.**Coursera iOS Specialization****Documentation and Guides:**Apple's Official Documentation:**Swift Documentation**iOS Developer Documentation**Ray Wenderlich's Tutorials:**In-depth tutorials and guides on iOS development.**Ray Wenderlich****GitHub Repositories:**Explore open-source iOS projects on GitHub to learn best practices and see real-world implementations.GitHub iOSBooks: "Swift Programming: The Big Nerd Ranch Guide" by Matthew Mathias and John Gallagher:A comprehensive guide to Swift programming suitable for beginners and experienced developers.**Swift Programming**"iOS 12 Programming for Beginners" by Ahmad Sahar:Geared towards beginners, this book covers iOS 12 development using Swift.iOS 12 Programming for Beginners**Community and Forums:**Stack Overflow - iOS Questions:Engage with the iOS developer community, ask questions, and learn from experienced developers.Stack Overflow iOSSwift.org Forums:Participate in discussions on Swift programming language and iOS development.Swift Forums |

**MODULES**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Scheduled Weeks** | **Module Title** | **Days** | **Hours** | **Learning Units** | **Tasks** |
| Week 1  |  Introduction, Getting Started with iOS 12 and Xcode 10  | Day 1 | Hour 1 | Download and Installing Xcode 10  | Task # 1,2 |
| Navigating Xcode 10  |
| Adding Objects to the Interface Builder  |
| Creating Actions and Outlets  |
| Coding in the Classes  |
| Using the iOS 12 Simulators  |
| Creating My First iOS 12 Application  |
| My First iOS 12 Application Source Code  |
| Building to a Real Device  |
|  Programming in Swift 4.2 - Learning the Basics 2hr 10min  | Hour 2 | Swift 4.2 Vars vs Lets - Variables vs Constants | Task # 3 |
| Swift 4.2 Strings  |
| Swift 4.2 Emojis  |
| Swift 4.2 Characters  |
| Swift 4.2 Ints - Integers  |
| Swift 4.2 Floats & Doubles - Decimals 32 & 64 bit |
| Swift 4.2 Booleans  |
| Swift 4.2 Arrays  |
| Hour 3 | Swift 4.2 Dictionaries  |
| Swift 4.2 Sets  |
| Swift 4.2 If Statements  |
| Swift 4.2 AND and OR Statements  |
| Swift 4.2 For Loops  |
| Hour 4 | Swift 4.2 While Loops  |
| Swift 4.2 Switch Statements  |
| Swift 4.2 Functions  |
| Swift 4.2 Optionals  |
|  Displaying Text within the Application  | Day 2 | Hour 1-2 | Displaying Text within the Application  | Task # 4 |
| Dismissing the Keyboard from the View  |
| Changing the Text Properties  |
| Displaying External Custom Fonts  |
| Hour 3-4 | iOS 12 APP: Fancy Text Creator Application |
| Fading and Revealing Objects  |
| Hiding and Revealing Objects  |
| Enabling and Disabling Objects  |
| Creating and Using NSTimers  |
| Displaying the Time and Date  |
| iOS 12 APP: Digital Clock Application  |
|  Swift 4.2 Reaction Based Games  | Day 3 | Hour 1-2 | iOS 12 APP: Traffic Lights Game Part 1 - Creating the Interface  | Task # 5 |
| iOS 12 APP: Traffic Lights Game Part 2 - Adding the Functionality |
| iOS 12 APP: Tap Me Fast Reaction Based Game Part 1 - Creating the Interface  |
| iOS 12 APP: Tap Me Fast Reaction Based Game Part 2 - Adding the Functionality |
|  Swift 4.2 Random Generators  | Hour 3-4 | Creating a Random Number Generator  | Task # 6 |
| iOS 12 APP: Roll of a Dice Application  |
| Creating a Random Word Generator  |
| iOS 12 APP: Random Facts Application  |
| Creating a Random PList Generator  |
| iOS 12 APP: Random Car Statistics Applications  |
|  Swift 4.2 Creating Actions  | Day 4 | Hour 1 | Creating a Action with a Button  | Task # 7 |
| Delaying Actions with a Timer  |
| Toggling Actions with UISwitches  |
| Using UISegmentedControls  |
| Controlling Actions with UISliders  |
| Displaying UIActionSheets  |
| Creating URL Links to Websites  |
|  Swift 4.2 Calculation Applications  | Hour 2 | iOS 12 APP: Calculator Application Part 1 - Creating the Interface  | Task # 8 |
| iOS 12 APP: Calculator Application Part 2 - Adding the Functionality |
| iOS 12 App: Temperature Calculator Part 1 - Creating the Interface  |
| iOS 12 App: Temperature Calculator Part 2 - Adding the Functionality  |
|  Swift 4.2 Gesture Based Games  | Hour 3-4 | Detecting Shake Gestures with the Device | Task # 9 |
| iOS 12 APP: Shake Me Silly Part 1 - Creating the Interface  |
| iOS 12 APP: Shake Me Silly Part 2 - Adding the Functionality |
| Detecting Swipe Gestures on the Screen  |
| iOS 12 APP: Simon Says Part 1 - Creating the Interface |
| iOS 12 APP: Simon Says Part 2 - Adding the Functionality |
|  Swift 4.2 Displaying Views  | Day 5 | Hour 1-2 | Using UIViews  | Task # 10 |
| Controlling UIScrollViews  |
| Displaying UIAlertViews  |
| Designing Interfaces with UIStackViews  |
| Switching to a New Storyboard View  |
| Switching to a New XIB View  |
| Changing the Status Bar Colour  |
| Removing the Status Bar  |
| Using UIPickerViews  |
| Using UIDatePickerViews  |
|  Swift 4.2 Web Browser Application  | Hour 3 | iOS 12 APP: Web View Application Part 1  | Task # 11 |
| iOS 12 APP: Web View Activity Indicator Part 2  |
| iOS 12 APP: Web View Search Bar Part 3  |
|  Swift 4.2 Map View Application  | Hour 4 | iOS 12 APP: Map View Application Part 1  | Task # 12 |
| iOS 12 APP: Map View Pins Part 2  |
| iOS 12 APP: Map View Locations Part 3  |
| iOS 12 APP: Map View Directions Part 4  |
| Week 2 |  Swift 4.2 Displaying Images  | Day 1 | Hour 1-2 | Displaying Images on the Screen  | Task # 13 |
| Displaying Retina Image Versions  |
| iOS 12 APP: Mini Image Gallery Application  |
| Adding Images To Buttons  |
| Displaying Images From URL'S  |
| Displaying Local PDF Files  |
| Changing the Background Colour  |
| Playing Animation Files  |
| Adding Borders to Objects  |
| Adding Drop Shadows to Objects  |
| Saving Images to the Device  |
| Creating Application Icons  |
| Adding Application Icons  |
| Creating a Universal Launch Screen  |
| Designing Loading Images  |
| Adding Loading Images  |
|  Swift 4.2 Collection Views Image Gallery  | Hour 2-4 | Populating Collection Views from an Array | Task # 14 |
| iOS 12 APP: Image Gallery Collection View PList Part 1  |
| iOS 12 APP: Image Gallery Collection View Cell Sizes Part 2  |
| iOS 12 APP: Image Gallery Collection View Pushing Part 3  |
| iOS 12 APP: Image Gallery Collection View Detail View Part 4  |
|  Swift 4.2 Camera Application  | Day 2 | Hour 1 | iOS 12 APP: Camera Application Using the Camera | Task # 15 |
| iOS 12 APP: Camera Application Using the Photo Library |
| iOS 12 APP: Camera Application Saving the Images  |
|  Swift 4.2 Snap Attack Image Game  | iOS 12 APP: Snap Attack Part 1  | Task # 16 |
| iOS 12 APP: Snap Attack Part 2  |
|  iOS 12 APP: Doodle Bug | Hour 2-4 | iOS 12 APP: Doodle Bug - Drawing on the Screen Part 1  | Task # 17 |
| iOS 12 APP: Doodle Bug - Preset Colours and Erase Part 2 |
| iOS 12 APP: Doodle Bug - Creating a Settings View Part 3  |
| iOS 12 APP: Doodle Bug - Changing the Brush Size Part 4  |
| iOS 12 APP: Doodle Bug - Adding Opacity Options Part 5 |
| iOS 12 APP: Doodle Bug - Saving the Artwork Part 6 |
| iOS 12 APP: Doodle Bug - Hiding the Buttons Part 7 |
|  Swift 4.2 Sticker Pack Application  | Day 3 | Hour 1-2 | iOS 12 APP: Creating Stickers Part 1  | Task # 18 |
| iOS 12 APP: Creating Animated Stickers Part 2 |
| iOS 12 APP: Adding Stickers to the Application Part 3  |
|  Swift 4.2 iMessage Application  | iOS 12 APP: Movie Quotes iMessage Application Part 1 | Task # 19 |
| iOS 12 APP: Movie Quotes iMessage Application Part 2 |
| iOS 12 APP: Movie Quotes iMessage Application Part 3 |
|  Swift 4.2 Audio & Video  | Hour 3-4 | Playing a Sound File  | Task # 20 |
| Adding Volume Controls  |
| iOS 12 APP: Animal Soundboard Application |
| Playing a Local Video File  |
| Streaming an Online Video  |
| Playing a YouTube Video  |
|  Swift 4.2 Table View Application + Revision  | Day 4 | Hour 1-4  | Populating a Table View from an Array  | Task # 21 |
| iOS 12 APP: London Landmarks - Populating a Table View Part 1 |
| iOS 12 APP: London Landmarks - Pushing to a Detail View Part 2  |
| iOS 12 APP: London Landmarks - Populating a Detail View Part 3  |
| iOS 12 APP: London Landmarks - Map View Locations Part 4 |
|  Swift 4.2 Tab Bar Business Application  | Day 5 | Hour 1-2 | iOS 12 APP: Business Application - Setting Up the Tab Views | Task # 22 |
| iOS 12 APP: Business Application - Creating the Services Pages  |
| iOS 12 APP: Business Application - Displaying the About Us |
| iOS 12 APP: Business Application - Creating a Portfolio Gallery |
| Hour 3-4 | iOS 12 APP: Business Application - Setting Up the Contact Us Page |
| iOS 12 APP: Business Application - In App Emailing  |
| iOS 12 APP: Business Application - Displaying Social Networks  |
| Week 3 |  Swift 4.2 Core Data Application  | Day 1 | Hour 1-2 | iOS 12 APP: Database - Creating a Database Part 1 | Task # 23 |
| iOS 12 APP: Database - Saving Data, Images and Text Part 2 |
| iOS 12 APP: Database - Editing and Updating Data Part 3 |
| iOS 12 APP: Database - Deleting Data Part 4 |
|  Swift 4.2 Social integration  | Hour 3 | Sending Text and Images via Email  | Task # 24 |
| Sending Text and Images via SMS Message  |
| Making a Phone Call  |
|  Swift 4.2 Secret Phonebook Application  | Hour 4 | iOS 12 APP: Secret Phone Book - Creating a Contact App Part 1 | Task # 25 |
| iOS 12 APP: Phone Book - Adding Contacts Part 2 |
| iOS 12 APP: Phone Book - Edit and Remove Contacts Part 3 |
| iOS 12 APP: Phone Book - Call and Message Contacts Part 4  |
|  Swift 4.2 Ad Revenue  | Day 2 | Hour 1 | Admob Banners Ads  | Task # 26 |
| Admob Interstitial Ads  |
| Admob Video Reward Ads  |
|  Swift 4.2 In App Purchases  | Hour 2-3 | iOS 12 APP: In App Purchases - Setting Up The App Part 1  | Task # 27 |
| iOS 12 APP: In App Purchases - Displaying The Content Part  |
| iOS 12 APP: In App Purchases - Purchasing The Content Part 3 |
| iOS 12 APP: In App Purchases - Saving The Purchase Part 4 |
| iOS 12 APP: In App Purchases - Restoring The Purchase Part 5  |
|  Swift 4.2 Universal Support  | Hour 4 | Device Detection  | Task # 28 |
| Screen Size Detection  |
| Different Device Storyboards  |
|  Learning Objective-C  | Day 3 | Hour 1 | Objective-C Variables and Constants  | Task # 29 |
| Objective-C Strings  |
| Objective-C Ints - Integers  |
| Objective-C Floats & Doubles - Decimals 32 & 64 bit |
| Objective-C Booleans  |
| Objective-C Arrays  |
| Objective-C If Statements  |
| Hour 2 | Objective-C AND and OR Statements  |
| Objective-C For Loops  |
| Objective-C While Loops  |
| Objective-C Switch Statements  |
| Objective-C Functions  |
|  Objective-C Creating Functions  | Hour 3 | Displaying Text within the Application  | Task # 30 |
| Dismissing the Keyboard from the View  |
| Changing the Text Properties  |
| Displaying External Custom Fonts  |
| Hour 4 | iOS 12 APP: Fancy Text Creator Application  |
| Fading and Revealing Objects  |
| Hiding and Revealing Objects  |
| Enabling and Disabling Objects  |
| iOS 12 APP: Digital Clock Application  | Day 4 | Hour 1 | Creating and Using NSTimers  | Task # 31 |
| Displaying the Time and Date  |
| iOS 12 APP: Digital Clock Application  |
|  Objective-C Reaction Based Games  | Hour 2 | iOS 12 APP: Traffic Lights Game Part 1  | Task # 32 |
| iOS 12 APP: Traffic Lights Game Part 2  |
| iOS 12 APP: Tap Me Fast Reaction Based Game Part 1 |
| iOS 12 APP: Tap Me Fast Reaction Based Game Part 2 |
|  Objective-C Random Generators  | Hour 3 | Creating a Random Number Generator  | Task # 33 |
| iOS 12 APP: Roll of a Dice Application  |
| Creating a Random Word Generator  |
| Hour 4 | iOS 12 APP: Random Facts Application  |
| Creating a Random PList Generator  |
| iOS 12 APP: Random Car Statistics Applications |
|  Objective-C Creating Actions  | Day 5 | Hour 1 | Creating a Action with a Button  | Task # 34 |
| Delaying Actions with a Timer  |
| Toggling Actions with UISwitches  |
| Using UISegmentedControls  |
| Controlling Actions with UISliders  |
| Displaying UIActionSheets  |
| Creating URL Links to Websites  |
|  Objective-C Calculation Applications  | Hour 2 | iOS 12 APP: Calculator Application Part 1 | Task # 35 |
| iOS 12 APP: Calculator Application Part 2 |
| iOS 12 App: Temperature Calculator Part 1 |
| iOS 12 App: Temperature Calculator Part 2 |
|  Objective-C Gesture Based Games  | Hour 3 | Detecting Shake Gestures with the Device  | Task # 36 |
| iOS 12 APP: Shake Me Silly Part 1  |
| iOS 12 APP: Shake Me Silly Part 2  |
| Detecting Swipe Gestures on the Screen  |
| Hour 4 | iOS 12 APP: Simon Says Part 1  |
| iOS 12 APP: Simon Says Part 2  |
| Week 4 |  Objective-C Displaying Views  | Day 1 | Hour 1 | Using UIViews  | Task # 37 |
| Controlling UIScrollViews  |
| Displaying UIAlertViews  |
| Designing Interfaces with UIStackViews  |
| Hour 2 | Switching to a New Storyboard View  |
| Switching to a New XIB View  |
| Changing the Status Bar Colour  |
| Removing the Status Bar  |
| Using UIPickerViews  |
| Using UIDatePickerViews  |
|  Objective-C Web Browser Application  | Hour 3 | iOS 12 APP: Web View Application Part 1  | Task # 38 |
| iOS 12 APP: Web View Activity Indicator Part 2  |
| iOS 12 APP: Web View Search Bar Part 3  |
|  Objective-C Map View Application  | iOS 12 APP: Map View Application Part 1  | Task # 39 |
| iOS 12 APP: Map View Pins Part 2  |
| iOS 12 APP: Map View Locations Part 3  |
| iOS 12 APP: Map View Directions Part 4  |
|  Objective-C Displaying Images  | Hour 4 | Displaying Images on the Screen  | Task # 40 |
| Displaying Retina Image Versions  |
| iOS 12 APP: Mini Image Gallery Application |
| Adding Images To Buttons  |
| Displaying Images From URL'S  |
| Displaying Local PDF Files  |
| Changing the Background Colour  |
| Objective-C Displaying Images  | Day 2 | Hour 1 | Playing Animation Files  | Task # 41 |
| Adding Borders to Objects  |
| Adding Drop Shadows to Objects  |
| Saving Images to the Device  |
| Creating Application Icons  |
| Adding Application Icons  |
| Creating a Universal Launch Screen  |
| Designing Loading Images  |
| Adding Loading Images  |
|  Objective-C Collection Views Image Gallery  | Hour 2-3 | Populating Collection Views from an Array | Task # 42 |
| iOS 12 APP: Image Gallery Collection View PList Part 1  |
| iOS 12 APP: Image Gallery Collection View Cell Sizes Part 2  |
| iOS 12 APP: Image Gallery Collection View Pushing Part 3 |
| iOS 12 APP: Image Gallery Collection View Detail View Part 4 |
|  Objective-C Camera Application  | Hour 3 | iOS 12 APP: Camera Application Using the Camera  | Task # 43 |
| iOS 12 APP: Camera Application Using the Photo Library  |
| iOS 12 APP: Camera Application Saving the Images |
|  Objective-C Snap Attack Image Game  | Hour 4 | iOS 12 APP: Snap Attack Part 1  | Task # 44 |
| iOS 12 APP: Snap Attack Part 2  |
|  Objective-C Drawing Application  | Day 3 | Hour 1 | iOS 12 APP: Doodle Bug - Drawing on the Screen Part 1  | Task # 45 |
| iOS 12 APP: Doodle Bug - Preset Colours and Erase Part 2  |
| iOS 12 APP: Doodle Bug - Creating a Settings View Part 3  |
|  Objective-C Drawing Application  | Hour 2 | iOS 12 APP: Doodle Bug - Changing the Brush Size Part 4  | Task # 44 |
| iOS 12 APP: Doodle Bug - Adding Opacity Options Part 5  |
| Hour 3 | iOS 12 APP: Doodle Bug - Saving the Artwork Part 6  |
| iOS 12 APP: Doodle Bug - Hiding the Buttons Part 7  |
|  Objective-C Sticker Pack Application  | iOS 12 APP: Creating Stickers Part 1  | Task # 45 |
| iOS 12 APP: Creating Animated Stickers Part 2  |
| iOS 12 APP: Adding Stickers to the Application Part 3  |
|  Objective-C iMessage Application  | iOS 12 APP: Movie Quotes iMessage Application Part 1  | Task # 46 |
| iOS 12 APP: Movie Quotes iMessage Application Part 2 |
| iOS 12 APP: Movie Quotes iMessage Application Part 3  |
|  Objective-C Audio & Video  | Hour 4 | Playing a Sound File  | Task # 47 |
| Adding Volume Controls  |
| iOS 12 APP: Animal Soundboard Application  |
| Playing a Local Video File  |
| Streaming an Online Video  |
| Playing a YouTube Video  |
|  Objective-C Table View Application  | Day 4 | Hour 1-2 | Populating a Table View from an Array  | Task # 48 |
| iOS 12 APP: London Landmarks - Populating a Table View Part 1  |
| iOS 12 APP: London Landmarks - Pushing to a Detail View Part 2  |
| iOS 12 APP: London Landmarks - Populating a Detail View Part 3  |
| iOS 12 APP: London Landmarks - Map View Locations Part 4  |
|  Objective-C Tab Bar Business Application  | Hour 2-4 | iOS 12 APP: Business Application - Setting Up the Tab Views | Task # 49 |
| iOS 12 APP: Business Application - Creating the Services Pages |
| iOS 12 APP: Business Application - Displaying the About Us  |
| iOS 12 APP: Business Application - Creating a Portfolio Gallery  |
| iOS 12 APP: Business Application - Setting Up the Contact Us Page  |
| iOS 12 APP: Business Application - In App Emailing  |
| iOS 12 APP: Business Application - Displaying Social Networks  |
|  Objective-C Core Data Application  | Day 5 | Hour 1 | iOS 12 APP: Database - Creating a Database Part 1  | Task # 50 |
| iOS 12 APP: Database - Saving Data, Images and Text Part 2  |
| iOS 12 APP: Database - Editing and Updating Data Part 3  |
| iOS 12 APP: Database - Deleting Data Part 4  |
|   |
|  Objective-C Social integration  | Sending Text and Images via Email  | Task # 51 |
| Sending Text and Images via SMS Message  |
| Making a Phone Call  |
|  Objective-C Secret Phonebook Application  | Hour 2 |   | Task # 52 |
| iOS 12 APP: Secret Phone Book - Creating a Contact App Part 1  |
| iOS 12 APP: Phone Book - Adding Contacts Part 2  |
| iOS 12 APP: Phone Book - Edit and Remove Contacts Part 3 |
| iOS 12 APP: Phone Book - Call and Message Contacts Part 4  |
|  Objective-C Ad Revenue  | Admob Banners Ads  | Task # 53 |
| Admob Interstitial Ads  |
| Admob Video Reward Ads  |
|  Objective-C In App Purchases  | Hour 3 | iOS 12 APP: In App Purchases - Setting Up The App Part 1  | Task # 54 |
| iOS 12 APP: In App Purchases - Displaying The Content Part 2  |
| iOS 12 APP: In App Purchases - Purchasing The Content Part 3  |
| iOS 12 APP: In App Purchases - Saving The Purchase Part 4  |
| iOS 12 APP: In App Purchases - Restoring The Purchase Part 5  |
|  Objective-C Universal Support  | Hour 4 | Device Detection  | Task # 55 |
| Screen Size Detection  |
| Different Device Storyboards  |

**Tasks for IoS Objective-C**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task No.** | **Task** | **Description** | **Week** |
| 1 | Introduction |  Set up development environment (install Xcode, create Apple ID). | Week 1 |
|  Familiarize yourself with the Xcode interface. |
|  Explore basic iOS concepts and architecture. |
| 2 | Getting Started with iOS 12 and Xcode 10 |  Download and install Xcode 10. |
|  Create a new Xcode project. |
|  Navigate through Xcode interface. |
|  Configure project settings. |
|  Understand the basics of Interface Builder. |
|  Add and arrange UI elements in Interface Builder. |
|  Connect UI elements to code using outlets and actions. |
|  Build a simple iOS app. |
| 3 | Programming in Swift 4.2 - Learning the Basics |  Learn about Swift variables and constants. |
|  Explore working with strings in Swift. |
|  Understand Swift data types, including integers and decimals. |
|  Use Swift booleans for decision making. |
|  Work with Swift arrays and dictionaries. |
|  Explore Swift sets. |
|  Implement if statements and logical operators. |
|  Create loops using for and while in Swift. |
|  Understand switch statements. |
|  Define and use functions in Swift. |
|  Explore Swift optionals. |
| 4 | Displaying Text within the Application |  Display static text in a UILabel. |
|  Dynamically update text content in response to events. |
|  Change text properties like font, color, and size. |
|  Use custom fonts in your iOS app. |
|  Create a simple text based iOS application. |
| 5 | Swift 4.2 Reaction-Based Games |  Create a traffic lights game with changing colors. |
|  Implement a tap me fast reaction based game. |
|  Design a digital clock application. |
|  Develop a reaction based game involving gestures. |
| 6 | Swift 4.2 Random Generators |  Generate random numbers in Swift. |
|  Make random choices using Swift. |
|  Shuffle elements in an array randomly. |
|  Create a rock, paper, scissors game with random choices. |
| 7 | Swift 4.2 Creating Actions |  Create actions and connect them to UI elements. |
|  Implement functionality for buttons and other controls. |
|  Handle user interactions with Swift code. |
| 8 | Swift 4.2 Calculation Applications |  Perform basic calculations using Swift. |
|  Build a calculator application. |
|  Explore advanced calculations with Swift. |
| 9 | Swift 4.2 Gesture-Based Games |  Implement gesture recognition in iOS apps. |
|  Develop games based on touch gestures. |
|  Enhance user interaction using gestures. |
| 10 | Swift 4.2 Displaying Views |  Understand the UIView class and its subclasses. |
|  Create custom views. |
|  Add and position views programmatically. |
|  Explore view animations and transitions. |
| 11 | Swift 4.2 Web Browser Application |  Integrate web views into your app. |
|  Create a simple web browser application. |
| 12 | Swift 4.2 Map View Application |  Integrate MapKit for displaying maps. |
|  Add annotations and overlays to maps. |
|  Develop a location based application. |
| 13 | Swift 4.2 Displaying Images |  Load and display images in iOS apps. | Week 2 |
|  Implement image caching for better performance. |
| 14 | Swift 4.2 Collection Views Image Gallery |  Use UICollectionView to create image galleries. |
|  Implement custom layouts for collection views. |
| 15 | Swift 4.2 Camera Application |  Access the device camera. |
|  Capture photos and videos. |
|  Build a simple camera application. |
| 16 | Swift 4.2 Snap Attack Image Game |  Create a game involving image snapping. |
|  Implement game logic based on images. |
| 17 | iOS 12 APP: Doodle Bug |  Develop a drawing application. |
|  Implement features like brush size and color selection. |
| 18 | Swift 4.2 Sticker Pack Application |  Create a sticker pack application. |
|  Allow users to send stickers in messaging apps. |
| 19 | Swift 4.2 iMessage Application |  Develop an iMessage application with interactive features. |
| 20 | Swift 4.2 Audio & Video |  Play audio and video in iOS apps. |
|  Explore multimedia capabilities of iOS. |
| 21 | Swift 4.2 Table View Application + Revision |  Implement table views for displaying lists of data. |
|  Perform revisions of Swift basics. |
| 22 | Swift 4.2 Tab Bar Business Application |  Create a tab bar |
|  based application. |
|  Design a business |
|  related app with multiple tabs. |
| 23 | Swift 4.2 Core Data Application |  Integrate Core Data for data persistence. | Week 3 |
|  Build an app with local data storage. |
| 24 | Swift 4.2 Social Integration |  Integrate social media sharing in your app. |
|  Implement features like sharing content on Twitter or Facebook. |
| 25 | Swift 4.2 Secret Phonebook Application |  Develop a secure phonebook application. |
|  Implement password protection for contacts. |
| 26 | Swift 4.2 Ad Revenue |  Integrate advertisements into your app. |
|  Explore strategies for ad revenue generation. |
| 27 | Swift 4.2 In App Purchases |  Implement in app purchase functionality. |
|  Add premium features or content for purchase. |
| 28 | Swift 4.2 Universal Support |  Ensure your app works across various iOS devices. |
|  Implement adaptive layouts for different screen sizes. |
| 29 | Learning Objective-C |  Familiarize yourself with Objective |
|  C syntax and basics. |
|  Create a simple iOS app using Objective C. |
| 30 | Objective-C Creating Functions |  Define and use functions in Objective C. |
|  Implement function based features in an app. |
| 31 | iOS 12 APP: Digital Clock Application |  Build a digital clock application. |
|  Display time and date in a user friendly format. |
| 32 | Objective-C Reaction-Based Games |  Develop reaction |
|  based games using Objective C. |
|  Implement interactive features based on user responses. |
| 33 | Objective-C Random Generators |  Generate random numbers and choices in Objective C. |
|  Apply randomness to game mechanics. |
| 34 | Objective-C Creating Actions |  Implement actions and connect them to UI elements in Objective C. |
|  Enhance app functionality through user interactions. |
| 35 | Objective-C Calculation Applications |  Perform calculations using Objective C. |
|  Develop a calculator application. |
| 36 | Objective-C Gesture-Based Games |  Utilize gestures for interactive gameplay in Objective C games. |
| 37 | Objective-C Displaying Views |  Create and manage views in Objective-C. | Week 4 |
|  Enhance user interfaces with custom views. |
| 38 | Objective-C Web Browser Application |  Develop a web browser application using Objective-C. |
| 39 | Objective-C Map View Application |  Integrate MapKit in Objective-C for map functionalities. |
| 40 | Objective-C Displaying Images |  Display images in iOS apps using Objective-C. |
| 41 | Objective-C Collection Views Image Gallery |  Implement image galleries using UICollectionView in Objective-C. |
| 42 | Objective-C Camera Application |  Access the device camera and implement basic photo capturing in Objective -C. |
| 43 | Objective-C Snap Attack Image Game |  Create an interactive image-based game using Objective-C. |
| 44 | Objective-C Drawing Application |  Develop a drawing application with basic drawing features in Objective -C. |
| 45 | Objective-C Sticker Pack Application |  Create a sticker pack application using Objective-C. |
| 46 | Objective-C iMessage Application |  Build an iMessage application with interactive elements using Objective-C. |
| 47 | Objective-C Audio & Video |  Implement audio and video playback in Objective-C. |
| 48 | Objective-C Table View Application: | Implement a UITableViewController in Objective-C. |
| Populate the table view with sample data. |
| Customize table view cells to display relevant information. |
| Handle row selection events and perform actions based on user interactions. |
| Explore different styles of table views (e.g., grouped, plain). |
| Implement search functionality within the table view. |
| Add a refresh control for updating table content. |
| Enhance the table view with custom sections and headers. |
| 49 | Objective-C Tab Bar Business Application: | Create a tab bar controller in Objective-C. |
| Design and implement multiple view controllers for each tab. |
| Customize tab bar icons and titles. |
| Implement navigation controllers within individual tabs for hierarchical content. |
| Explore modal presentation for specific tasks or information. |
| Utilize the delegate pattern for communication between tab bar items. |
| Implement features relevant to a business application (e.g., product catalog, customer information, analytics). |
| Add a settings tab for user preferences. |
| 50 | Objective-C Core Data Application: | Integrate Core Data framework into an Objective-C project. |
| Define data models and entities using Core Data. |
| Implement basic CRUD (Create, Read, Update, Delete) operations with Core Data. |
| Display data from Core Data in table views or collection views. |
| Set up relationships between different entities in Core Data. |
| Implement data validation and error handling. |
| Explore Core Data fetch requests for querying data. |
| Add sorting and filtering options for Core Data queries. |
| 51 | Objective-C Social Integration: | Implement social media sharing features using Objective-C. |
| Integrate social media SDKs (e.g., Facebook, Twitter) into the project. |
| Allow users to share content from the app to their social media accounts. |
| Implement authentication for social media accounts. |
| Fetch and display social media feeds within the app. |
| Utilize social media APIs for additional functionalities. |
| 52 | Objective-C Secret Phonebook Application: | Develop a secure phonebook application using Objective-C. |
| Implement user authentication for accessing the phonebook. |
| Encrypt sensitive contact information stored in the app. |
| Allow users to add, edit, and delete contacts securely. |
| Implement password recovery or reset mechanisms. |
| Add additional security features like biometric authentication (if supported). |
| Ensure data privacy and compliance with relevant regulations. |
| 53 | Objective-C Ad Revenue: | Integrate ad networks (e.g., AdMob, Facebook Audience Network) into the Objective-C project. |
| Display banner ads within specific view controllers. |
| Implement interstitial or rewarded ads for user engagement. |
| Track ad impressions, clicks, and revenue using ad network SDKs. |
| Explore mediation to maximize ad revenue from multiple networks. |
| Optimize ad placements for better user experience and engagement. |
| 54 | Objective-C In-App Purchases: | Implement in-app purchase functionality using Objective-C. |
| Define and configure products for purchase within the App Store Connect. |
| Handle transactions and receipts securely. |
| Implement consumable and non-consumable in-app purchases. |
| Provide a seamless purchasing experience for users. |
| Implement a restore purchases feature. |
| Ensure compliance with App Store guidelines for in-app purchases. |
| 55 | Objective-C Universal Support: | Ensure the app's user interface adapts to different screen sizes and orientations. |
| Implement adaptive layouts using size classes. |
| Use Auto Layout to create responsive interfaces. |
| Test the app on various iOS devices and simulators. |
| Optimize assets for different resolutions (e.g., images for iPhone and iPad). |
| Implement universal design principles for a consistent user experience. |
| Handle device-specific features and capabilities programmatically. |
| Optimize performance for both iPhones and iPads. |

**Equipment Required:**

* You will need a Apple Mac computer or laptop
* You will need the developer software Xcode 10

# *Annexure-II:*

# Motivational Lectures

**A Complete iOS 12 and Xcode 10 Course with Swift 4.2 & Objective-C**

**Apple Certified iOS Developer:**

Offered by Apple, this certification validates your proficiency in iOS development. However, Apple's certifications generally cover the broader aspects of iOS development and may not be version-specific.

**Apple Certified iOS Developer**

**Coursera Specializations:**

Coursera offers specializations that cover iOS development, Swift, and related topics. Courses from institutions like the University of Toronto and others can provide a comprehensive learning path.

**Coursera iOS Specialization**

**LinkedIn Learning Certifications:**

LinkedIn Learning offers various courses on iOS development, Swift, and related technologies. While they may not issue specific certifications, completing courses can enhance your LinkedIn profile and showcase your skills.

LinkedIn Learning iOS Development Courses

**Udacity Nanodegree Program: iOS Developer Nanodegree:**

Udacity offers a nanodegree program in iOS development, covering essential skills and technologies. Completing the program earns you a nanodegree credential.

**Udacity iOS Developer Nanodegree**

**Ray Wenderlich's iOS Path:**

Ray Wenderlich's online platform offers an extensive path for iOS development. While it may not issue certifications, completing the path demonstrates a broad understanding of iOS development.

**Ray Wenderlich Learning Paths**

**Annexure-IV:**

 **Workplace/Institute Ethics Guide**

Work ethic is a standard of conduct and values for job performance. The modern definition of what constitutes good work ethics often varies.  Different businesses have different expectations. Work ethic is a belief that hard work and diligence have a moral benefit and an inherent ability, virtue, or value to strengthen character and individual abilities. It is a set of values-centered on the importance of work and manifested by determination or desire to work hard.

The following ten work ethics are defined as essential for student success:

1. **Attendance:**Be at work every day possible, plan your absences don’t abuse leave time. Be punctual every day.
2. **Character:**Honesty is the single most important factor having a direct bearing on the final success of an individual, corporation, or product. Complete assigned tasks correctly and promptly. Look to improve your skills.
3. **Team Work:**

The ability to get along with others including those you don’t necessarily like. The ability to carry your weight and help others who are struggling. Recognize when to speak up with an idea and when to compromise by blend ideas together.

1. **Appearance:**Dress for success set your best foot forward, personal hygiene, good manner, remember that the first impression of who you are can last a lifetime
2. **Attitude:**Listen to suggestions and be positive, accept responsibility. If you make a mistake, admit it. Values workplace safety rules and precautions for personal and co-worker safety. Avoids unnecessary risks. Willing to learn new processes, systems, and procedures in light of changing responsibilities.
3. **Productivity:**Do the work correctly, quality and timelines are prized. Get along with fellows, cooperation is the key to productivity. Help out whenever asked, do extra without being asked. Take pride in your work, do things the best you know-how. Eagerly focuses energy on accomplishing tasks, also referred to as demonstrating ownership. Takes pride in work.
4. **Organizational Skills:**

Make an effort to improve, learn ways to better yourself. Time management; utilize time and resources to get the most out of both. Take an appropriate approach to social interactions at work. Maintains focus on work responsibilities.

1. **Communication:**Written communication, being able to correctly write reports and memos.
Verbal communications,being able to communicate one on one or to a group.
2. **Cooperation:**Follow institute rules and regulations, learn and follow expectations. Get along with fellows, cooperation is the key to productivity. Able to welcome and adapt to changing work situations and the application of new or different skills.
3. **Respect:**Work hard, work to the best of your ability. Carry out orders, do what’s asked the first time. Show respect, accept, and acknowledge an individual’s talents and knowledge. Respects diversity in the workplace, including showing due respect for different perspectives, opinions, and suggestions.