

## iOS: Final Project

### A. Custom Video Camera App

- Functional Requirements: Build an app that lets you do the following
  - take both regular and slow-motion video on an iPhone
  - save them to your own custom video library in the app's Documents directory
  - browse through your video library
  - play/stop/ your videos
- Technical Requirements
  - **UI Layer (Wireframes): Create simple wireframes using Balsamiq or any other wireframing tool.**
  - **Create a logical model on how/where your files/thumbnails will be stored and what naming conventions you will use.**
  - Browse your video library using the UICollectionView. You'll need to extract thumbnails from the videos.
  - Use asynchronous programming patterns using blocks.
- Create a presentation on how you designed and implemented the overall system
- Think about these questions
  - How would you store the videos in the cloud? How would you make sure user A does not see user B's photos?
- Reference:  
<https://developer.apple.com/library/ios/documentation/AudioVideo/Conceptual/AVFoundationPG/AVFoundationPG.pdf>

### B. Programming Exercises

1. Write a program that takes in a number and prints it in words. e.g. for 201, it should print "two hundred and one". Support up to billions.
2. Convert a decimal input to binary. e.g. for 224, it should print 11100000 - without using any built in functions
3. Convert a string to a number without using built-in functions e.g for "3.1415" it should convert it to 3.1415 and for "123" it should convert to 123.
4. Sort exercise
  - Read the following paper  
<https://drive.google.com/file/d/0BxL9QaRYkQyceFJnbG9fa25HN0E/view?usp=sharing>
  - Make sure you fully understand all the details and formulas presented in the paper
  - Now implement your own code for insertion sort
  - Start with sorting N=100000 (or more) randomly generated numbers and print the time spent in sorting. Then change N (double it, triple it and so on) and print the time each time. It should approximately match the analysis in the paper.
  - Generate sufficient data and plot a graph (N on X axis and time on Y axis) using Google Spreadsheet or Excel and make sure your graph matches the Big-O formula. Your goal should be that you run the program once and it generates a CSV file for you.
  - Note: You'll need to choose N correctly so you can see variations in time when you change N. That depends on how fast your CPU is.