**Programming Assignment # 5 due 03/19/2018**

Fragments are smaller pieces of an activity that are parts of a grouping of a views and functionalities. Fragments allow views to be bundled together and to be held within an activity. For example, in a portrait view, a single list of items appear on one screen and a detail view of each item as a detail screen on small screen devices; whereas on a tablet both the list and the detail views appear on the same screen.

In this assignment, you will create an app that will list all the computer science courses and the catalog detail description of each course using fragments. The list of courses is to be created in the user interface at runtime (dynamic UI).The courses and their descriptions are taken from the university catalog and are given below toward the end and also provided as a text file named Lab5.txt. When your start the New Android Studio Project, you must use the following:

Folder Name: Lab5

Application name and project: CS Courses

Activity Name: MainActivity (Entry point activity)

Layout Name: layout/activity_main and layout-land/activity_main (xml for the entry activity)

Name the class that extends ListFragment as CourseFragment, the class that extends Fragment as CourseDetails, the class that extends Activity as CourseActivity, and the course information class as Courses. You are not allowed to use a different set of Java class names for this assignment, if you choose another set of names you may lose some points. Please make sure that your work will run on a lab computer before you submit it for grading. Sample display using Nexus 6 API 26: initial display in portrait mode is Figure 1, the same in landscape mode is Figure 2. When CS 4391 is clicked the display in portrait mode is Figure 3, and the same in landscape mode is Figure 4.

Please note that you are not required to have the same colors as in the figures. If you don't like them please feel free to try a different set of colors but not the same colors as in the class demo app. If you have time research more and get pretty displays.


Figure 1 Initial display in portrait mode

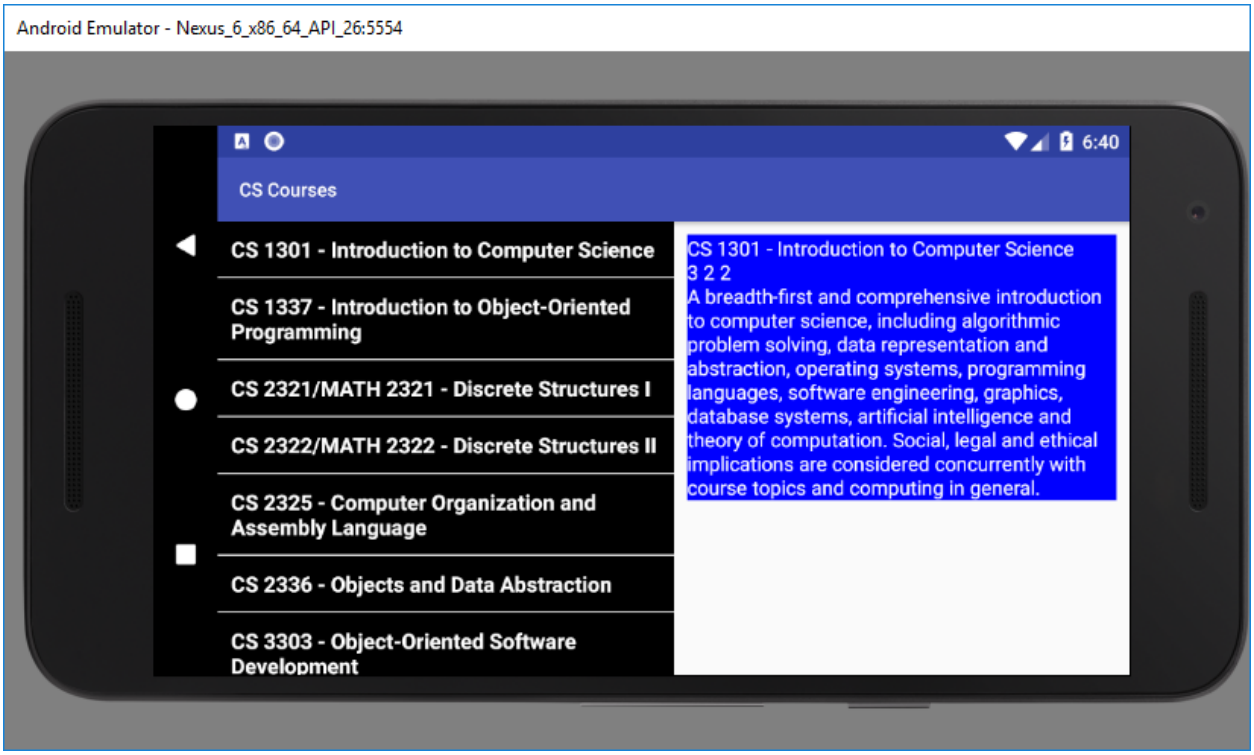


Figure 2 Initial display in landscape mode

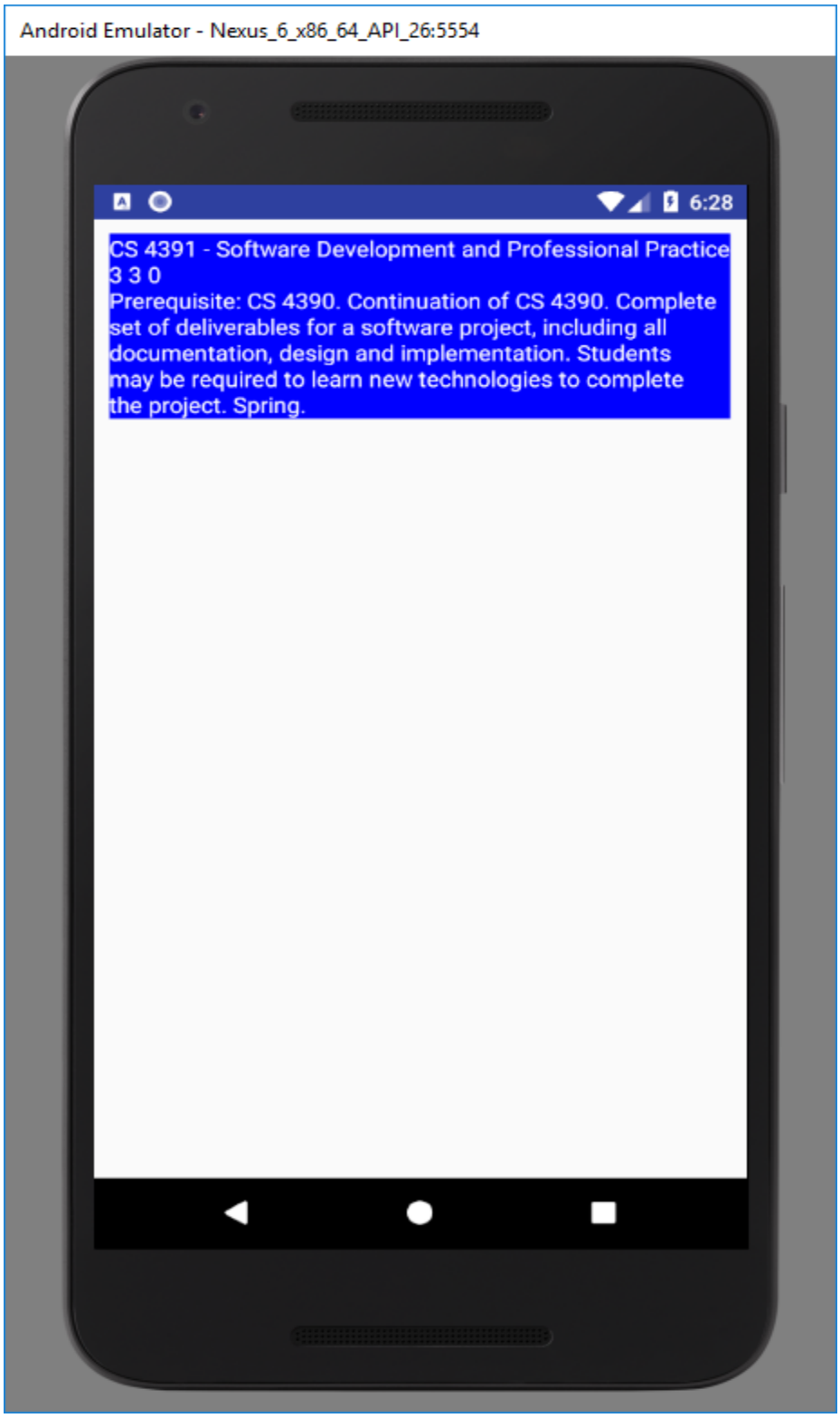


Figure 3 - CS 4391 course description in portrait mode

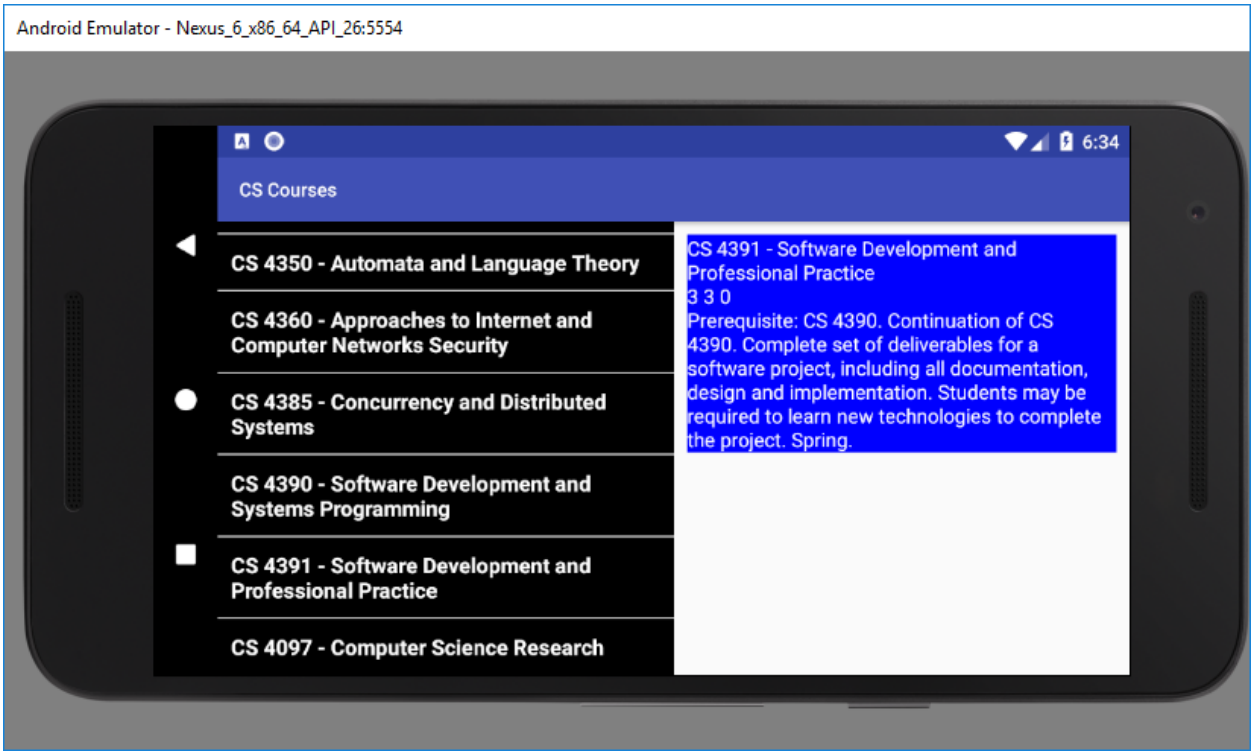


Figure 4 -- CS 4391 course description in landscape mode

The list of the CS courses and the descriptions:

CS 1301 - Introduction to Computer Science

CS 1337 - Introduction to Object-Oriented Programming

CS 2321/MATH 2321 - Discrete Structures I

CS 2322/MATH 2322 - Discrete Structures II

CS 2325 - Computer Organization and Assembly Language

CS 2336 - Objects and Data Abstraction

CS 3303 - Object-Oriented Software Development

CS 3305 - Data Structures and Algorithms

CS 3307 - Algorithm Design and Analysis

CS 3310 - Programming Languages

CS 3315 - Scripting Languages

CS 3321 - Introduction to Enterprise Systems

CS 3322 - Enterprise Systems Application Development

CS 3352 - Operating Systems and Networking

CS 3372 - Net-Centric Computing

CS 3387 - Artificial Intelligence

CS 4321 - Enterprise Systems Assembler Programming

CS 4322 - Advanced Topics of Enterprise Systems

CS 4325 - Computer Architecture

CS 4330 - Computer Graphics

CS 4340 - Database Systems Use, Design and Implementation

CS 4350 - Automata and Language Theory

CS 4360 - Approaches to Internet and Computer Networks Security

CS 4385 - Concurrency and Distributed Systems

CS 4390 - Software Development and Systems Programming

CS 4391 - Software Development and Professional Practice

CS 4097 - Computer Science Research

The course descriptions:

CS 1301 - Introduction to Computer Science

3 2 2

A breadth-first and comprehensive introduction to computer science, including algorithmic problem solving, data representation and abstraction, operating systems, programming languages, software engineering, graphics, database systems, artificial intelligence and theory of computation. Social, legal and ethical implications are considered concurrently with course topics and computing in general.

CS 1337 - Introduction to Object-Oriented Programming

3 3 2

Fundamental concepts of programming from an object-oriented perspective, including simple data types, control structures, array and string data structures, algorithms, debugging techniques and the social implications of computing. Emphasis of good software engineering principles. Fall, spring, summer.

CS 2321/MATH 2321 - Discrete Structures I

3 3 0

Prerequisite: MATH 1316 or MATH 2412. Introduction of the foundations of discrete mathematics as they apply to computer science, focusing on providing a solid theoretical foundation for further work. Topics include functions, relations, sets, simple proof techniques, Boolean algebra, propositional logic, digital logic, elementary number theory and the fundamentals of counting. Spring.

CS 2322/MATH 2322 - Discrete Structures II

3 3 0

Prerequisite: CS 2321 or MATH 2321. Continuation of the discussion of discrete mathematics introduced in CS 2321 or MATH 2321. Topics include predicate logic, recurrence relations, graphs, trees, matrices, computational complexity, elementary computability and discrete probability. Spring.

CS 2325 - Computer Organization and Assembly Language

[COSC 2325]

3 2 2

Prerequisite: CS 2336 and CS 2321 or MATH 2321. Introduction to the internal operation of modern computer systems. Topics include low-level hardware details (combinational and sequential circuits), data representation (number systems, character encoding, integer floating point representation), basic

computer system organization (CPU, I/O, memory and disk), and assemblers and assembly language. Course fee $25.

CS 2336 - Objects and Data Abstraction

3 2 2

Prerequisite: CS 1337. Methodology of programming from an object-oriented perspective. Through the study of object-oriented design, an introduction to the basics of human-computer interfaces, graphics and the social implications of computing, with an emphasis on software engineering.

CS 3303 - Object-Oriented Software Development

3 3 0

Prerequisite: CS 2336 or consent of instructor. Software engineering techniques for programming in the large, intermediate issues in object-oriented programming language design patterns, frameworks and systems support software. Course fee $25.

CS 3305 - Data Structures and Algorithms

3 3 0

Prerequisite: CS 2336, CS 2321 or MATH 2321 and MATH 2413 or consent of computer science coordinator. Continuation of the study of data abstraction begun in CS 2336. Topics include recursion, the underlying philosophy of object-oriented programming, (including stacks, queues, linked lists, hash tables, trees and graphs), basics of algorithmic analysis and introduction to principles of language translation. Course fee $25.

CS 3307 - Algorithm Design and Analysis

3 3 0

Prerequisite: CS 2322 or MATH 2322 and CS 3305. Introduces formal techniques to support the design and analysis of algorithms, focusing on both the underlying mathematical theory and practical considerations of efficiency. Topics include asymptotic complexity bounds, techniques of analysis, algorithmic strategies, and an introduction to automata theory and its application to language translation.

CS 3310 - Programming Languages

3 3 0

Prerequisite: CS 3305. Introduction to the main constructs of contemporary programming languages with elaboration on the tools necessary for critical evaluation of existing and future programming languages. Course fee $25. Fall.

CS 3315 - Scripting Languages

3 3 0

Prerequisite: CS 2336 or consent of instructor. Introduces the fundamentals of operating systems use and administration. Topics include the use of editors, file systems, processes, TCP/IP and the Internet, regular expressions, filters and system and network administration using various scripting languages. Applications of these concepts will be realized using shell scripting and other scripting languages. Course fee $25.

CS 3321 - Introduction to Enterprise Systems

3 3 0

Prerequisite: CS 2336 or consent of computer science coordinator. Introduction to concepts of mainframe and enterprise class computers including usage and architecture. Topics will include fundamentals of z/OS® operating system, mainframe workloads and major middleware applications in

use on mainframes today, and introduction to systems and application programming. Course fee $25. Fall.

CS 3322 - Enterprise Systems Application Development

3 3 0

Prerequisite: CS 3321 or consent of computer science coordinator. Course is intended for mainframe application developers who wish to use workstation-based tools to build applications for enterprise systems. Students will be introduced to use of IBM Rational Developer for System z. Course fee $25. Spring.

CS 3352 - Operating Systems and Networking

3 3 0

Prerequisite: CS 3305. Fundamentals of operating systems design and implementation. Topics include an overview of the components of an operating system, mutual exclusion and synchronization, implementation of processes, scheduling algorithms, memory management, and file systems. Basics of networking and communications and their impact on operating systems. Course fee $25.

CS 3372 - Net-Centric Computing

3 3 0

Prerequisite: CS 3352. Introduction of the structure, implementation and theoretical underpinnings of computer networking and the applications that have been enabled by the technology. Fall.

CS 3387 - Artificial Intelligence

3 3 0

Prerequisite: CS 3305. Introduction of fundamental concepts and techniques of artificial intelligence.

CS 4321 - Enterprise Systems Assembler Programming

3 3 0

Prerequisite: CS 3322 or consent of computer science coordinator. Intermediate-level course for application and system programmers and others who wish to write systems and application programs in Assembler Language. Appropriate introduction for individuals wishing to deepen insight into System z architecture. Course fee $25. Fall.

CS 4322 - Advanced Topics of Enterprise Systems

3 3 0

Prerequisite: CS 4321 or consent of computer science coordinator. Advanced topics of relevance to enterprise systems. Course fee $25. Spring.

CS 4325 - Computer Architecture

3 3 0

Prerequisite: CS 2322 or MATH 2322, CS 2325 and CS 3305. Review of von Neumann architecture and its limitations; parallel computer structures and concurrent computation; pipeline computers and vectorization methods; array processors; multiprocessor architectures and programming; dataflow computers.

CS 4330 - Computer Graphics

3 3 0

Prerequisite: CS 3305. Principles of interactive computer graphics; systems organization and device technologies for raster and vector displays; 2-D and 3-D viewing, clipping, segmentation and interaction handling 3-D geometrical transformations, projections and hierarchical data structures for graphics modeling.

CS 4340 - Database Systems Use, Design and Implementation

3 3 0

Prerequisite: CS 3305. Focus on underlying design and performance of database systems including relational and object-oriented systems. Fall.

CS 4350 - Automata and Language Theory

3 3 0

Prerequisite: CS 3307, CS 3310 and senior standing. Introduction to theoretical aspects of programming languages. Finite automata, context-free grammars, Turing machines, Chomsky hierarchy and Church-Turing thesis.

CS 4360 - Approaches to Internet and Computer Networks Security

3 3 0

Prerequisite: CS 3372. Presentations of essential computer network security in wired and wireless network topologies, including authentication, encryption and server side protection. Packets, TCP/IP,

UDP, ports, Internet topology, IP addressing scheme and DNS. Hands-on security breeches will be presented and practices implementing security in a local area network (LAN). Course fee $25.

CS 4385 - Concurrency and Distributed Systems

3 3 0

Prerequisite: CS 3372. Introduces concurrency in the context of distributed systems. The course covers both the abstract principles of concurrent programming and their concrete realization in distributed, network-based systems. Topics include the basic theory of concurrency, hardware and software features to support concurrency, concurrent and distributed algorithms, and middleware. Course fee $25.

CS 4390 - Software Development and Systems Programming

3 3 0

Prerequisite: senior standing and consent of computer science coordinator. Extends the ideas of software design and development to encompass problems encountered in medium- and large-scale systems. Software engineering, professionalism, ethical responsibilities in software development and human-computer interaction. Fall.

CS 4391 - Software Development and Professional Practice

3 3 0

Prerequisite: CS 4390. Continuation of CS 4390. Complete set of deliverables for a software project, including all documentation, design and implementation. Students may be required to learn new technologies to complete the project. Spring.

CS 4097 - Computer Science Research

1-6 0 0

Prerequisite: consent of department head. Selected individual research topics in computer science to accommodate more substantial research than is encountered in normal course work. May be repeated for a maximum of six hours.

**Steps to turn in your assignment:**

(1) This project is an individual submission project.
(2) Copy the app folder Lab5 into the ~/Documents/cs3303/gitRep/XX99999 directory, where XX999999 is your user_id.
(3) cd into XX99999 directory and enter the command git add Lab5
(4) enter the command git commit –m "My lab 5 work mm/dd/yy"
(5) enter the command git push origin master
(6) To test that the folder has been pushed successfully, go to your gitRepTest folder and enter the command git pull. If you get back a copy of the folder Lab5 and its content then your submission has been pushed successfully. However, you need to verify that your submission is what you want your instructor to grade.