INTERMEDIATE ANDROID DEVELOPMENT
Course Syllabus

Course Number: APD-0248
OCAS Code: None
Course Length: 120 Hours
Career Cluster: Information Technology
Career Pathway: Programming and Software Development
Career Major(s): Mobile Development

Pre-requisite(s): Fundamentals of Technology or Business and Computer Fundamentals

Course Description: This course in Intermediate Android Development covers the concepts of Android use and its development. Students download and install the tools necessary for developing Android applications. Students create application manifests, applications that use external resources to provide dynamic support for locations, languages, and hardware configurations. Students enhance views and layouts, create Activities and Broadcast events, connect to Internet resources, and save and manage preferences for the end user. Students build databases, and expand usability with Content Providers, create services that run independently, and build menus and navigation that enhance the user’s experience. Students design Apps that use the Text-to-Speech and speech recognition libraries and use animations. Students program the device for sensory input and screen orientations. Students design for interactive Maps, build Widgets, add audio and video capabilities, integrate face and feature recognition, manage Bluetooth, NFC, Network, and WI-FI capabilities as well as SMS messaging, and Android Beaming. Students also learn how to sign an application for distribution, publish the application, as well as promote and monetize it.

iTunes U

Course Objectives:

A. Demonstrate Knowledge of Android Tools
   1. Define Android and explain the background of mobile application development
   2. Download and install the SDK for Android development
   3. Identify and verbalize which devices run on Android

B. Demonstrate Knowledge of Beginning Android Programming
   1. Create an environment for Android development and for debugging projects including designing for small screens and mobile data connections.
   2. Explain the importance of optimizing for speed and efficiency.
   3. Manipulate Android Virtual Devices, the Emulator and other development tools for use in android program development.

C. Demonstrate Knowledge of Creating Applications and Activities
   1. Identify and explain the Android application components and different types of applications that can be built with them.
   2. Build an application manifest.
   3. Design external resources to provide dynamic support for locations, languages, and hardware configurations.
4. Design and use a self-created Application class to expand usability of a device
5. Construct new Activities that complete a life cycle and end

D. Demonstrate Knowledge of Building User Interfaces
1. Apply and optimize Views and Layouts correctly for the end user
2. Build code that will Extend, Group, Create, and use Views accurately on a device
3. Utilize and manipulate Adapters to bind data to Views for a device

E. Demonstrate Knowledge of Intents and Broadcast Receivers
1. Define and describe the use of Intents for programs
2. Create Starting Activities, sub-Activities and Services using implicit and explicit Intents for a program
3. Build code that "Broadcasts" events using the Broadcast Intents on a device
4. Construct and evaluate a Pending Intent for different programs
5. Verbalize how to use Intent filters and Broadcast Receivers when switching programs
6. Modify code to "listen" to Broadcast Intents, monitor device state changes, and manage manifest receivers at run time on a device

F. Demonstrate Knowledge of Using Internet Resources
1. Apply code to Connect to Internet resources for a device
2. Construct code to Parse XML resources for a device
3. Produce code that will use and Query the Download Manager on a device
4. Manipulate the Account Manager to authenticate with Google App Engine in tandem with a device

G. Demonstrate Knowledge of Files, Saving State and Preferences
1. Parse simple application data using Shared Preferences on a device
2. Save Activity instance data between sessions for the end user
3. Manage application preferences and build Preference Screens on a device
4. Save and load files and manage the local file system for a device
5. Manipulate static files to Include as external resources for a device

H. Demonstrate Knowledge of Databases and Content Providers
1. Create databases and use SQLite on a device
2. Manipulate Content Providers, Cursors, and Content Values to store, share and consume applications data on a device
3. Asynchronously query Content Providers using Cursor loaders on a device
4. Add Search capabilities to an application for a device
5. Develop code that will use the native Media Store, Contacts, and Calendar Content Providers for a device

I. Demonstrate Knowledge of Working in the Background
1. Create, Start, and Stop Services for Android devices
2. Build code to Bind Services to Activities on a device
3. Develop Ongoing foreground services on a device
4. Extend the Intent Service for an application on a device
5. Manipulate the AsynchTasks to manage background processing on a device
6. Construct background Threads and use Handlers to synchronize with the GUI Thread on a device
7. Apply and manipulate code to use Alarms to schedule application events for a device

J. Demonstrate Knowledge of Expanding the User Experience
1. Create and manipulate code to customize the Action Bar on a device
2. Manipulate the Action Bar for application navigation on a device
3. Manipulate the Android menu system for adding code to enhance a device’s usefulness
4. Evaluate and manipulate Action bar actions to enhance the use of the device
5. Utilize and manipulate immersive applications on a device
6. Create and display Dialogs for the end user
7. Manipulate the Notification Manager to notify users of application events on a device
8. Create Insistent and ongoing Notifications for the end user

K. Demonstrate Knowledge of Advanced User Experience
1. Build Resolution Independence and design for every screen to enhance visibility on a device
2. Create image assets in XML to enhance usability of a device
3. Develop applications accessible to the end user
4. Design code that uses the Text-to-Speech and speech recognition libraries to extend the use of the device for audio instructions
5. Utilize animations for visual appeal on the device
6. Manipulate hardware acceleration to allow for different program deliveries on a device
7. Create and manipulate Surface Views on a device
8. Utilize code that uses the copy and paste options, and the clipboard for end users to manipulate content

L. Demonstrate Knowledge of Hardware Sensors
1. Manipulate the Sensor Manager for hardware sensors on a device
2. Define and describe the available sensor types for hardware sensors
3. Write code that finds a device’s natural orientation and remap a device’s orientation reference frame
4. Manipulate the code for sensors to monitor a device’s environment

M. Demonstrate Knowledge of Maps, Geocoding, and Location-Based Services
1. Define and explain forward and reverse geocoding as it applies to device maps
2. Develop interactive maps with Map Views and Map Activities for interaction with the end user
3. Create and add Overlays to device maps
4. Manipulate code to find a location with location-based services for the end user
5. Build and manipulate proximity alerts for the end user

N. Demonstrate Knowledge of Invading the Home Screen
1. Create home screen Widgets for a device
2. Develop collection-based home screen Widgets for a device
3. Manipulate code to use Content Providers to populate Widgets for a device
4. Design and manipulate surface search results to the Quick Search box for the end user

O. Demonstrate Knowledge of Audio, Video, and Using the Camera
1. Construct code to play audio and video with the Media Player for the end user
2. Build and manipulate audio focus and media button presses for the end user
3. Use the Remote Control Client to control external devices
4. Produce code to Record audio and video with the Media Recorder for the end user
5. Create code to Record video and taking pictures using Intents on the device
6. Design code to Preview recorded video and display live camera streams for the end user
7. Assemble code for the end user to manipulate raw audio
8. Generate code to utilize face and feature recognition for end users
P. **Demonstrate Knowledge of Bluetooth, NFC, Networks and Wi-Fi**
   1. Write code to manage Bluetooth devices and discoverability mode
   2. Construct code to discover remote Bluetooth devices
   3. Design code to communicate over Bluetooth between devices
   4. Build code to monitor a device’s Internet connectivity
   5. Generate code to monitor a device’s Wi-Fi and network details
   6. Design code to transfer data using Wi-Fi Direct
   7. Assemble code to scan NFC tags on the device
   8. Build code to transferring data between devices using Android Beam

Q. **Demonstrate Knowledge of Telephony and SMS**
   1. Assemble code to initiate phone bills on the device
   2. Construct code to read the phone, network, data connectivity, and SIM states on the device
   3. Generate code to monitor changes to the phone, network, data connectivity, and SIM states on the device
   4. Write code to use Intents to send SMS and MMS messages to and from the device
   5. Generate code to manipulate the SMS Manager to send SMS messages to and from the device
   6. Build code to handle incoming SMS messages for the device

R. **Demonstrate Knowledge of Advanced Android Development**
   1. Develop code to secure Android using permissions
   2. Write code to send server pushes with Cloud to Device Messaging
   3. Construct code to add copy protection with the License Verification Library on the device
   4. Build code to monetize with In-App Billing on the device
   5. Assemble code to use Wake Locks on the device
   6. Construct code to inter-process communication (IPC) using AIDL and Parcelables on the device
   7. Write code to improve the device’s application performance using Strict Mode
   8. Assemble code to ensure the device’s backward and forward hardware and software compatibility

S. **Demonstrate Knowledge of Monetizing, Promoting, and Distributing Applications**
   1. Construct code to create a signing certificate for the device
   2. Write code to sign applications for distribution
   3. Build code to publish on Google Play
   4. Design code to monetize strategies for the device
   5. Develop code for promotion strategies on the device
   6. Assemble code to use Google Analytics for the end user

**ODCTE Objectives**

**Teaching Methods:** The class will primarily be taught by the lecture and demonstration method and supported by various media materials to address various learning styles. There will be question and answer sessions over material covered in lecture and media presentations. Supervised lab time is provided for students to complete required projects.
INTERMEDIATE ANDROID DEVELOPMENT

Grading Procedures:
1. Students are graded on theory and shop practice and performance.
2. Each course must be passed with seventy (70%) percent or better.
3. Grading scale: A=90-100%, B=80-89%, C=70-79%, D=60-69%, F=50-59%.

Description of Classroom, Laboratories, and Equipment:
Tulsa Technology Center campuses are owned and operated by Tulsa Technology Center School District No. 18. All programs provide students the opportunity to work with professionally certified instructors in modern, well-equipped facilities.

Available Certifications/College Credit
The student may be eligible to take state, national or industry exam after completion of the program. College credit may be issued from Oklahoma State University-Okmulgee or Tulsa Community College. See program counselor for additional information.

College Credit Eligibility:
The student must maintain a grade point average of 2.0 or better.