Develop and publish your own original apps for Android™ mobile devices!

Students use state of the art development tools to design apps that can have an impact in their communities. They create apps using Java and XML that can fully utilize all capabilities of mobile devices.

Computer Science A (CSA) builds on the basic skills learned in Computer Science Principles (CSP) to teach students authentic Android™ app development. Students in this course continue to hone their communication and collaboration skills while learning to use a variety of tools. The primary goal of the course is to create independent-thinking app developers; every unit in this course builds on students' prior knowledge and skills until they are able to complete an app development cycle independently from the ground up.

PLTW's CSA is designed to cover all learning objectives in the College Board's AP Computer Science A framework, and exceeds the College Board's requirement of 20 hours of lab activity. It is also an example of a CSTA level 3C course. Activities, projects, and problems are provided to the teacher via the Professional Development tool in the form of student-ready handouts, teacher notes, and supplementary materials including code and slide presentations.

The course is designed to be readily adaptable to student interests and community assets. Individual teachers are encouraged to modify the course content so that it feels as authentic and meaningful within the local context as possible. This course aims to fully develop Object Oriented Programming (OOP) skills that were introduced in CSP and will require consummate engagement with the material for success. As such, augmenting content to keep it fresh and exciting is a priority. The following is a summary of the units of study that are included in the course. Please note there are two versions of Unit 4, which are detailed later in this document.

CSA Unit Summary

Unit 1	Introducing Java (22%)
Unit 2	Vanilla Android™ Development (25%)
Unit 3	Advanced Android™ Features (28%)
Unit 4 (AP Focus)	Games and Algorithms (25%)
-or-	
Unit 4 (Android Focus)	The LibGDX Game Development Framework (25%)

Unit 1: Introducing Java

Unit one provides a primer in the basics of the Java programming language and object oriented programming (OOP). Students create classes, instantiate them, add instance data and access that data. They use conditionals, iteration, arithmetic and logical operators, arrays and iterators, first in BlueJ to ensure code correctness, and then in Android™ Studio to incorporate their own code into fully functional apps. The material provided also includes extra practice on all these Java topics and more.

Lesson 1.1	Objects in Java
Lesson 1.2	Manipulating Data

Lesson 1.1: Objects in Java

The goal of this lesson is to give students the tools they need to create Java objects. They will manage data by creating their own methods and calling them. They'll add the logic into an app that notifies the user of appropriate courses of action when they head out the door in the morning based on the weather forecast. The lesson concludes with students augmenting the artificially intelligent natural language processing MagPie app based on the College Board's MagPie chatbot lab.

Activity 1.1.1	Introduction to Android™ Development (2 days)
Activity 1.1.2	Your First Class (3 days)
Activity 1.1.3	Making Objects (4 days)
Activity 1.1.4	If It's Raining... (5 days)
Activity 1.1.5	Your Sci-fi Name (1 day)
Project 1.1.6	Chatting with Magpie (7 days)

Lesson 1.2: Manipulating Data

This lesson focuses on the management of data in arrays and use of iteration in Java. Students will write code that parses string data and plug that code into an app that retrieves data from the device's SD card. They also write code to manage and maintain a list of billboard ratings. The unit culminates with students creating an app from the ground up that uses buttons to play sound assets.

Activity 1.2.1	Parsing Text (4 days)
Activity 1.2.2	Today's Top 40 (5 days)
Activity 1.2.3	Data Storage (5 days)
Project 1.2.4	Create an Android Project (1 day)
Problem 1.2.5	Synthesizer (2 days)

Unit 2: Vanilla Android™ Development

Students will spend most of this unit developing out a viewer for college applications, with college admissions officials as the target audience. Highlights of this unit include working with fragments, mastering encapsulation, and designing and implementing apps that utilize the most common and useful user interface elements. Students will use a Backend As A Service (BaaS) to implement persistent data within their app, allowing a user to access their data from any Android™ device. At the end of the unit, students create a new app, applying what they have learned throughout the unit.

Lesson 2.1	App Navigation
Lesson 2.2	Data Persistence

Lesson 2.1 App Navigation

A sample of the final product to be produced in the next few lessons is provided. Students begin this lesson by testing out its functionality. The College App is designed to quickly show an admissions officer whatever assets an applicant has provided in a mobile format. Students learn to incorporate and extend several common User Interface (UI) features. In the process they learn about inheritance and class definitions as well as transferring their prior knowledge of basic Python language constructs to Java. To add real feeling and usability to their apps, students will incorporate navigation features, including menus and action bars. Students explore and critique a Unified Modeling Language UML diagram and other documentation for the College App.

Activity 2.1.1	Usability Testing (1 day)
Project 2.1.2	Prototyping with proto.io (2 days)
Activity 2.1.3	Classes (1 day)
Activity 2.1.4	App Navigation (3 days)
Project 2.1.5	User Input (4 days)

Lesson 2.2 Data Persistence

This lesson continues to emphasize the OOP paradigm, reinforcing previous learning. Additionally, students learn about and use some common data structures including ArrayLists. Students create their own checked exceptions, and access a BaaS to implement data persistence. They make classes that inherit from interfaces or other classes and use these within data structures, necessitating a solid understanding of polymorphism. They Google's libraries for taking and displaying pictures in the College App. Having completed the College App, students will have fully utilized the ArrayList class in one dimension, extended interfaces and abstract classes, overloaded methods and more. Using all of the app-making skills, they will create a new app based on what they have learned from their completed College App.

Activity 2.2.1	Exceptions and Scope (1 day)
Activity 2.2.2	Remote Database (4 days)
Activity 2.2.3	ListView (4 days)
Activity 2.2.4	One Method, Many Classes (4 days)
Activity 2.2.5	List and Detail (2 days)
Activity 2.2.6	Let Me Take a Selfie (3 days)
Problem 2.2.7	The Development Process (7 days)

Unit 3: Advanced Android™ Features

The goal of Unit Three is for students to reach a level of understanding of Google's Android™ libraries that allow them to create apps that use a broad range of mobile features such as Global Positioning Systems (GPS) and web access. The major project in this unit is a social networking app that utilizes the Google Cloud Platform to store data. Students begin by learning about the Google Cloud Datastore service and using client and server code to read and write to a cloud database. Students then learn to access the GPS features of their mobile devices and move on to reading Quick Response (QR) codes and accessing the web. The unit culminates in a problem in which students create a geo-cache style app using the techniques they developed in the social networking app projects.

Lesson 3.1	Trip Tracker
Lesson 3.2	Location Awareness
Lesson 3.3	Contacts in an App
Lesson 3.4	App Analysis

Lesson 3.1 Trip Tracker

In this lesson, students will learn to add a backend service to their apps. Students will store and retrieve user data from the cloud. This provides teachers with an opportunity to connect some of the OOP that students have been learning in Java to authentic web frameworks and APIs. The goal of this lesson is to give students the power to create apps that store significant amounts of data, that can benefit from crowd/cloud-sourcing, and that allow users to access data from anywhere. In the second half of this lesson, students will begin constructing a basic interface for a social networking app including prototyping and usability testing. This app will be enhanced throughout this unit.

Activity 3.1.1	Trip Tracker Startup (2 days)
Activity 3.1.2	User Authentication (2 days)
Activity 3.1.3	A New Trip (2 day)
Activity 3.1.4	Listing Trips (1 days)
Activity 3.1.5	Updates and Deletes (3 days)
Activity 3.1.6	Public vs. Private Trips (2 days)
Activity 3.1.7	Sort Algorithms (3 days)
Activity 3.1.8	Search Algorithms (2 days)
Project 3.1.9	Social Media Networking App – Design (2 days)
Project 3.1.10	Social Media Networking App – Development (5 days)

Lesson 3.2 Location Awareness

In this lesson students will reinforce their learning of basic Java language constructs and OOP while making their apps location aware. They will then add location data to posts that users make.

Activity 3.2.1	Preparing for Google Play Services (1 day)
Activity 3.2.2	Using Google Play Services (2 days)
Project 3.2.3	Location Awareness – Design (1 day)
Project 3.2.4	Location Awareness – Development (4 days)

Lesson 3.3 Contacts in an App
This lesson concludes work on the Trip Tracker app by adding the ability to query the device to discover locally stored contacts. Each contact is categorized into a subclass of an abstract type, and then polymorphed into a large list. The finished product will require students to have used common features such as autoboxing, common methods from the String class, and dynamic late binding, in addition to reinforcing all of the Java constructs from the previous unit. Students create a Geo-Quest app which combines the camera feature from Unit 2, the geolocation feature from Lesson 3.2 and knowledge of polymorphism, lists, and strings. The app keeps track of a list of polymorphed quest items to find around campus; when a team member discovers a quest item, the user will use their app to record it with a geolocation and an image of the item. When all required items have been found, the quest is complete.

Activity 3.3.1	Trip Cost and Rating (3 days)
Activity 3.3.2	Polymorphic People (2 days)
Activity 3.3.3	Persistent People (2 days)
Activity 3.3.4	Polymorphic Behavior (3 or more days)
Problem 3.3.5	GeoQuest (5 days)

Lesson 3.4 App Analysis
The goal of this lesson is to explore nuances of the Java language, especially those that occur during computational algorithms. Students will analyze the performance of various sorts and searches, perform statement execution counts, learn a simple rounding algorithm, experiment with operator precedence, witness integer overflow, and convert between the hexadecimal and decimal number systems.

Activity 3.4.1	Investigating Sort (1 day)
Activity 3.4.2	Investigating Search (1 day)
Activity 3.4.3	Computations in Java (2 days)

Unit 4: The LibGDX Game Development Framework (Android Focus)
The goal of Unit 4 is to give students an opportunity to practice and refine their understanding of Java techniques in the context of game development. LibGDX is a popular open source game development framework that is constantly growing due to the contributions of its active community members. An important part of this lesson is teaching students to access resources to help themselves utilize all the tools that are available to them. Students learn to incorporate media assets, and work with graphics and touch events. Students access and manipulate data in 2-D data structures, and interpret code created using the MVC pattern before making significant modifications of their own. Finally, students create a unique app that incorporates elements like geolocation, communication with a database, and utilization of the camera, speakers, and microphone. The choice of app theme and topic are left to the student, though they should target a specific audience, and benefit their community in some way.
Students need to find a “client” with whom they communicate regularly about the progress of the project. This could be the manager of a GitHub repository, a community leader, or a local business owner. Students might also choose to develop a full-fledged game at this point in the course with their fellow students as the clients.

Lesson 4.1	Creating a New World
Lesson 4.2	Graphic Adventure Game
Lesson 4.3	Independent Projects

Lesson 4.1 Creating a New World
The goal of this lesson is to get students to understand the foundations of game development in LibGDX. The end product uses the touch screen to register user input. Students incorporate 2-D graphic assets into the project and manipulate 2-D data structures.

Activity 4.1.1	LibGDX Setup (1 day)
Activity 4.1.2	Level Loading (2 days)
Activity 4.1.3	Walls, Characters, and Doodads (5 days)

Lesson 4.2 Graphic Adventure Game
In this lesson, students transfer their knowledge to fix problems with existing source code and add entirely new features to an existing game. In order to meaningfully improve the existing code, students must use math and problem-solving skills as well as Java and object-oriented concepts covered previously in this course.

Activity 4.2.1	Code Overview (1 day)
Activity 4.2.2	Erratic Movement (1 day)
Project 4.2.3	Game Improvement (5 days)

Lesson 4.3 Independent Projects
Before the start of this lesson, students may opt to investigate bonus activities that cover animation outside of LibGDX and publish apps they develop. In this lesson students put into practice everything they've explored in the course thus far. Likely more than for any task they've taken on up to this point, this problem requires students to be effective teammates, collaborators, communicators, and developers. Students are tasked with finding an authentic "client" for their work with whom regular communication is embedded in the Agile design process. This project may be a continuation of student work from Unit 2. Student projects should address authentic needs in their community, though the choice of problem is left to the individual students. Example projects might include a water conservation awareness app for clients living in the drought-riddled Southwest, or a puzzle game for friends to play. Students might choose to make an app that allows students to register online for courses at their school, develop an educational game for younger students, or perhaps even build an interactive set of tutorials for this very course. The possibilities are limitless, but the time available is not. Students need to practice expert time management skills in order to create successful apps.

Problem 4.3.1	Make an App (20 days)

AP Unit 4: The LibGDX Game Development Framework (AP Focus)
The purpose of AP Lesson 4.1 and Lesson 4.2 is to focus more on AP concepts and not on Android™ app development. If you are not delivering the course with a focus on AP and/or your students would rather develop Android apps, you may want to teach the complimentary Unit 4: The LibGDX Game Development Framework.
In this lesson, students return to the BlueJ environment to learn two-dimensional (2D) arrays and their algorithms. They will enhance a game of Concentration and use array algorithms to manage the 2D array of game cards. The game will also use abstract classes and inheritance.

AP Lesson 4.1	The Game of Concentration
AP Lesson 4.2	Elevens
AP Lesson 4.3	Integrate and Create

AP Lesson 4.1 The Game of Concentration
Lesson 4.1 introduces two-dimensional arrays and nested looping in a game called Concentration. You will create a simple text-based version of the game, explore various array algorithms, and modify the game to create a variant of a game called Sevens.

AP Activity 4.1.1	Two-Dimensional Arrays (1 day)
AP Activity 4.1.2	Concentration (4 days)

AP Lesson 4.2 Elevens
The activities in this lesson are related to a simple solitaire game called Elevens. You will learn the rules of Elevens, which are similar to the rules of Sevens from the last lesson. You will be able to play it by using the supplied User Interface (UI) shown below. You will learn about the design and the object-oriented principles that suggested the design. You will also implement much of the code.

AP Activity 4.2.1	Design and Create a Card Class (1 day)
AP Activity 4.2.2	Design a Deck Class (2 days)
AP Activity 4.2.3	Create Shuffle Algorithms (2 days)
AP Activity 4.2.4	Add a Shuffle Method to Deck (2 days)
AP Activity 4.2.5	Test with Simulations (Optional) (2 days)
AP Activity 4.2.6	Playing Elevens (1 day)
AP Activity 4.2.7	Design the Elevens Board (1 day)
AP Activity 4.2.8	Using an Abstract Board Class (2 days)
AP Project 4.2.9	Implement Elevens (3 days)

AP Lesson 4.3 Integrate and Create
For their final project, students will have the choice to continue working with their Elevens project, integrating new features and creating a new implementation of their Concentration game, or they may choose to return to the Android Development environment and create an app described in *Android Problem 4.3.1 Make and App*.

In this Integrate and Create project, students will analyze all aspects of the Elevens program, make slight modifications to create a game of Thirteens, and then make major modifications to make a graphical version of their game of Concentration. Learning, analyzing, and modifying existing code is in-demand skill that students will begin to master in this lesson.

It is also a fun project to do after the AP Test, and introduces no new concepts.

Problem 4.3.1	Make an App (15 days)