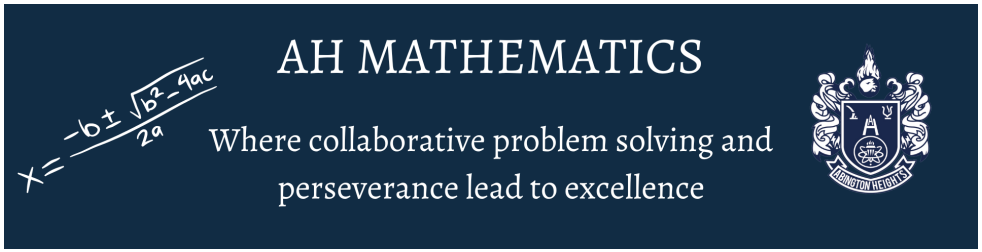
Abington Heights School District
Computer Science III Honors:
Mobile App Development Curriculum

In Computer Science III Honors: Mobile App Development, students develop their computer programming skills through the following areas of study:

1. User Interface Design
2. Event-Driven Programming
3. Navigation and Lists
4. App Architecture Principles
5. Networking

Board Approval Date:
Adoption: 2023 - 2024 SY
Review Date:



Abington Heights Math Framework

Stakeholders	Actions
Students	★ Engage in mathematical discussions, share their ideas openly, be inquisitive, seek to understand and learn more about mathematical concepts, and try their best daily. ★ Exhibit creativity and curiosity in problem solving individually and collaboratively. ★ Persevere in engaging and challenging daily mathematical practice. ★ Come prepared to learn every day.
Teachers	★ Create a safe and collaborative classroom environment where students feel vested in a shared vision for mathematical excellence. ★ Develop high quality instruction that meets the needs of all learners through differentiation. ★ Use a variety of 21st century methodologies to advance learning. ★ Partner with parents and guardians to support student success. ★ Establish a collaborative community within the building and amongst grade levels to ensure a cohesive level of instruction.
Building Leaders	★ Deeply understand the needs of teachers, students, the instructional materials being used, programs being implemented, and the expectations for state-level assessment scores ○ Knowledgeable about program and grade level standards ○ Ensure consistent and equal access to high-quality instructional materials and resources, building. ★ Be partners with teachers, students and families: ○ Provide guidance and support to the mathematical community. ○ Understand needs of teachers, students and families. ★ Trust the educators to make professional decisions based on program, student, and district needs.
Central Admin	★ Effectively communicate to the school board and community specific areas of need and how to support teachers and building leaders in a quest for mathematical excellence ★ Deeply understand the needs of teachers, students, the instructional materials being used, programs being implemented, and the expectations for state-level assessment scores ○ Have a common metric for mathematical excellence. ○ Ensure consistent and equal access to high-quality instructional materials and resources, district. ○ Re-examine best practices/curriculum routinely (6 years). ★ Support a culture of collaboration between the other stakeholder groups to maintain the standard of excellence of the Abington Heights ★ Trust the educators to make professional decisions based on program, student, and district needs.
Parents/ Community	★ Be a strong support system and contribute by building a positive math community for students. ★ Encourage a positive math mindset. ★ Have conversations with their children about school and ask what they are learning about in school. ★ Be open, receptive to the district's ideas about student learning and reach out to teachers/school to learn more about how they can support. ★ Trust the educators to make professional decisions based on program, student, and district needs.
School Board	★ Provide the fiscal resources to support: ○ Highly qualified professionals for mathematics ○ High-quality instructional materials ○ Effective and efficient math interventions for remediation ○ Professional development for math content and instructional practices ★ Trust the educators to make professional decisions based on program, student, and district needs.

Computer Science III Honors: Mobile App Development Scope and Sequence

Month	Unit	Estimated Number of Weeks
September	Kotlin	2 1/2
	Introduction To Android	1
October	User Interface	3
	Interactive Apps	1
November	Interactive Apps	3
	View Binding With Activities	1/2
December	View Binding With Activities	1
	Activity Lifecycle & Log Messages	1
	Introduction To Fragments	1
January	Introduction To Fragments	1/2
	Navigation Component	3 1/2
February	Navigation Component	1 1/2
	Recycler View	2 1/2
March	Recycler View	1
	Viewmodel And LiveData	2
	Networking	1
April	Networking	2 1/2
	Final Project	1 1/2
May	Final Project	4
June	Final Project	1/2

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
KOTLIN	What are the advantages of using Kotlin? When should you use "var" vs "val"? How does casting differ from Java? How do you compare help values in Kotlin? When is the "else" required at the end of a when statement? What are the differences between arrays, lists and mutable lists in Kotlin? What functions can you use on a nullable variable? What is the syntax for declaring a Kotlin function?	The programming language of Kotlin Creating and running a Kotlin project Mathematical operators Mutable variables, immutable variables and constants Casting Strings and String templates Comparing values Conditionals - If/else statements - When statements Arrays, Lists and MutableLists For Loops and Ranges Nullability ——————————— Functions - Named parameters - Default values - Compact functions	Run Kotlin code in IntelliJ Use Kotlin's math functions to perform math operations Declare mutable / immutable variables with var / val Print variables and Strings using a String template Use a conditional within a String template Use a when statement inside of an if/else Use a when statement within a print statement Use a when statement to assign a variable Declare an array / list / mutable list Declare a mixed, nested array Add or set elements within a mutable list Write a for loop with ranges to print Declare, assign and reassign a null variable	Set up IntelliJ for Kotlin Introduction to Kotlin slides - Coding "Try Its" - Code tracing throughout slides Kotlin Basics programming lab Kotlin Functions, Lambdas and Classes slides - Coding "Try Its"	Q&A during slides Coding "Try Its" throughout slides Code tracing throughout slides Kotlin Basics programming lab

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
KOTLIN (continued)	What are higher order functions and how are lambdas useful for them? Where do you initialize properties within a class? When is the init block called? What symbol is used to extend a class or implement an interface?	Lambda expressions Classes - Primary constructor - Class properties - init block - Getters and setters - lateinit variables Inheritance and Interfaces Data Classes	Declare and implement a function with parameters and a return value - with and without default values Write a lambda expression and store it in a variable Write a custom getter / setter Declare a lateinit variable Define a type by declaring a data class; create an instance of that type		
INTRODUCTION TO ANDROID	What are the advantages of Android being open source? What are the challenges of Android app development? How should you choose a minimum SDK version for your project? What steps do you	Android as open source Android Development Kit (ADK) AndroidStudio as IDE ——————————— API levels Navigating Android Studio - Menu on left - MainActivity and activity_main.xml - Project folders - Android manifest - java folder - res folder - Gradle	Create a new project in AndroidStudio Locate the tool within AndroidStudio that helps you pick a minimum SDK Find the MainActivity and activity_main.xml files Create a new virtual test device for the emulator Run your app on the emulator Run your app on a physical device	Android Introduction slides Your First Android app slides Creating a HelloWorld app (activity) Running the HelloWorld app on the emulator and devices (activity)	Q&A during slides Creating a HelloWorld app (activity) Running the HelloWorld app on the emulator and devices (activity)

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
INTRODUCTION TO ANDROID (continued)	have to take on a physical phone to set it up for use as a test device?				
USER INTERFACE **USER**	What are the essential attributes all views require? What is the importance of density - independent pixels? What folder do you add downloaded images to? ——————— Why are ViewGroups necessary? What is the orientation of a LinearLayout by default? What's the difference between margins and paddings? ——————— What is the	XML - Elements - Closing tags - Attributes View size - dp - Width and height - Text size View identifiers Useful TextView attributes Other common views ImageViews - Scale types - Adding your own image ——————— ViewGroups LinearLayout Layout Weight Gravity Margins and Padding View Ids in XML	Set the width and height of a TextView using hardcoded dp and wrap_content Set the text size for a TextView Change background color of TextView using hex code Add a View to a layout (EditText, Button, CheckBox, RadioButton, SeekBar, Spinner, ImageView) Add a downloaded image to your project using the drawables folder Set the scaletype of an ImageView ——————— Create a LinearLayout with multiple views within it Change the orientation of a LinearLayout to vertical Use layout_weight to equally space views Use gravity to center text horizontally and/or vertically Use padding to add space between a	Views and XML slides Views and XML discovery programming lab Views and XML quiz LinearLayout slides - Coding "Try Its" Linear Layout discovery programming lab LinearLayout quiz LayoutEditor and ConstraintLayout slides - Coding "Try Its" User Interface programming lab ConstraintLayout	Q&A during slides Coding "Try Its" throughout slides Views and XML discovery programming lab Views and XML quiz Linear Layout discovery programming lab LinearLayout quiz User Interface programming lab Constraint Layout quiz Documentation

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
INTERFACE (continued)	advantage of using ConstraintLayout? What is the minimum number of constraints a View must have? What value for height/width doesn't exist in ConstraintLayout? What do you use instead?	ScrollView ——————————— LayoutEditor tool ConstraintLayout - Creating constraints - Aligning sides - Aligning baselines - Adding a margin - Using guidelines - Bias Sizes in ConstraintLayout ConstraintLayout toolbar Chaining views - Types of chains ——————————— *(Documentation Day)* Styles and Themes	view and its contents Use margins to add space between a view and its parent Declare an id for a view in XML Wrap a view in a ScrollView ——————————— Add a custom font to a project Create views using the Palette and Component Tree Set view attributes using the Attributes panel Constraint a view to the parent view, another view, or aligned with another view Set a margin on a constraint Align the baseline of views Create a guideline and use it as a constraint Adjust a view's bias Chain multiple views together ——————————— *(Documentation Day)* Create a style and theme within a project and use it for your xml	quiz Documentation Day programming activity	Day programming activity

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
INTERACTIVE APPS	What is an activity? What is event driven programming? How do you use a resource id within XML vs Kotlin? Why must you call findViewById after onCreate has been called? What function do you override to implement a click listener? ——————— Can you change the minimum and/or maximum value of a SeekBar? What are the three functions you must implement for a SeekBar listener? Where do you declare the values for a Spinner's dropdown menu?	Defining an Activity Activity's onCreate function Resource ids Inflation of views findViewById function Referencing view ids in Kotlin code Event Driven Programming and Listeners Responding to RadioButtons being pressed Responding to Checkboxes being selected String resources Toasts ——————— SeekBar - max attribute - progress SeekBar's listener - Its 3 required functions	Use the findViewById function within an activity to find and inflate a view Assign the view returned from findViewById to a variable Assign a Click Listener to a view Assign one Click Listener to multiple views Override the onClick function to to respond to a button press Override the onClick function by writing a lambda (with and without the view argument) Determine what view was pressed by using the view argument's id Define a RadioGroup and add a click listener to it Use the isChecked function to determine if a Checkbox was selected Create a String resource to define a new String Define a String resource for an existing String Display a Toast ———————————	Interactive Apps slides - Coding "Try Its" Interactive Apps programming lab Interactive Apps quiz Documentation Day programming activity 1 SeekBars and Spinners slides - Coding "Try Its" SeekBars and Spinners programming lab SeekBars and Spinners quiz Documentation Day programming activity 2	Q&A during slides Coding "Try Its" throughout slides Interactive Apps programming lab Interactive Apps quiz Documentation Day programming activity 1 SeekBars and Spinners programming lab SeekBars and Spinners quiz Documentation Day programming activity 2

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
INTERACTIVE APPS (continued)	What are the two functions you must implement for a AdapterView listener? ——————— How are Snackbars different from Toasts?	Spinner - Dropdown menu values - SpinnerAdapter / ArrayAdapter AdapterView's listener - Its 2 required functions ——————— (*Documentation Day 1)* Snackbars (*Documentation Day 2)* App Icons	Get the SeekBar's max and progress attributes programmatically Set the SeekBar's max and progress attributes programmatically Display a toast upon a SeekBar's progress changing Define an array of spinner values in strings.xml Create an ArrayAdapter using the createFromResource function Link an adapter instance to a Spinner instance Display a Toast when the user selected an option from the Spinner ——————— *(Documentation Day 1)* Display a Snackbar when the user presses a button (*Documentation Day 2)* Create a custom app icon for an existing app		
VIEW BINDING WITH ACTIVITIES	What are the advantages of using View Binding over findViewById?	Downsides of using findViewById Introduction to View Binding	Enable view binding within the project's gradle file Create a binding object within an activity	ViewBinding with Activities slides - Code-along demo app throughout slides	Q&A during slides Code-along demo app

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
VIEW BINDING WITH ACTIVITIES (continued)	What step creates the binding classes? Within what file do you enable view binding? What views automatically get properties created for them in the binding class? Using the binding object, how do you access the parent view in the corresponding layout file? Do you need properties for views once you have view binding? ——————— What's the difference between explicit and implicit intents?	Binding classes - Naming scheme for binding classes - Naming scheme for view properties within binding classes Enabling view binding Creating and initializing a binding object Binding object's root view How setContentView changes How to convert findViewById calls to instead use view binding ——————— (*Documentation Day)* Implicit Intents	Initialize a binding object within an activity using the inflate function Access the binding object's root view Update setContentView argument to use view binding Access views within the activity using the binding object Convert an existing app to use view binding Create a new app that uses view binding from the start ——————— (*Documentation Day)* Use implicit intents to allow a user to make a call to a prefilled number from within an app Use implicit intents to allow a user to email a prefilled address from within an app Use implicit intents to open the maps application and show a prefilled location Use implicit intents to open a website at a prefilled web addresses	ViewBinding programming lab ViewBinding with Activities quiz Documentation Day programming activity	ViewBinding programming lab ViewBinding with Activities quiz Documentation Day programming activity

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
ACTIVITY LIFECYCLE & LOG MESSAGES	Who's responsible for calling the lifecycle methods? What lifecycle methods are called when you… - First run an app? - Press the back button? - Press the recents button? - Press the home button? - Start the app from Recents? - Lock the device? - Unlock the device? - Rotate the device? Why does onCreate get called when you rotate a device? What view states does the system save by default?	Lifecycle states Lifecycle callback methods - onCreate - onStart - onResume - onPause - onStop - onDestroy The Log class Logcat output Back button Recents button Home button onRestart function Device configuration Landscape layout files Saving data with onSaveInstanceState ——————————— (*Documentation Day)* Debugging	Override activity lifecycle methods Use the Log class to display a log message in the console Choose the appropriate log level and use the corresponding function Use the dropdown menu and filters within Logcat to find a log message Create a resource directory for landscape layouts Create a layout file for an activity when its orientation is "landscape" Override on the onSaveInstanceState function Save a value by using the appropriate bundle function for that value's data type Declare a constant Retrieve saved data using the appropriate bundle "get" function ——————————— (*Documentation Day)* Use the debugging tools within AndroidStudio to find bugs in an app	Activity Lifecycle and Logging slides - Code-along demo app throughout slides Activity Lifecycle and Logging programming lab Activity Lifecycle and Logging quiz Documentation Day programming activity	Q&A during slides Code-along demo app Activity Lifecycle and Logging programming lab Activity Lifecycle and Logging quiz Documentation Day programming activity

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
INTRODUCTION TO FRAGMENTS	What are the advantages of fragments? How is a fragment's lifecycle affected by the host activity's lifecycle? What is the benefit to using the single activity architecture? What function within the default fragment code should you keep? What attributes should you set for the fragment container view? How do we know the get only binding property won't be null?	What is a Fragment? Host activity Fragment's lifecycle method "onCreateView" Single activity architecture Creating a Fragment UI Fragments ViewBinding with Fragments Get-only properties Converting an app to single activity architecture	Create a new fragment within an AndroidStudio project Determine and keep only the necessary code that is automatically generated when you create a fragment Move layout code from an activity layout file to a fragment layout file Create a container view for the fragment within the activity's layout file Declare a nullable binding property Declare a get only property for the binding object Reset the binding object within onDestroyView Duplicate an AndroidStudio project Convert an existing app to use fragments	Introduction to Fragments slides - Code-along demo app throughout slides Introduction to Fragments programming lab Introduction to Fragments quiz	Q&A during slides Code-along demo app Introduction to Fragments programming lab Introduction to Fragments quiz

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
NAVIGATION COMPONENT	What are the three parts of the Navigation Component? Which of the three do you create yourself? Which of the three manages navigation by swapping destinations in and out? Which of the three is the resource file that contains all navigation paths? Which of the three is a container that displays the fragment? What classes are automatically generated when you add SafeArgs?	Destinations, actions and navigation paths NavigationGraph - In navigation editor - In code - Adding destinations - Anatomy of a destination - Set the start destination - Adding actions - Adding arguments NavHost - NavHostFragment - NavHost attributes NavController - findNavController Passing information between screens using SafeArgs - Origination destinations - Receiving destinations - action functions - Retrieving arguments The backstack The navigateUp function	Edit the project's gradle file to enable navigation Create a navigation graph resource file Add destinations to the navigation graph within the navigation editor Destinate a screen as the start destination Connect destinations by creating actions in the navigation editor Add a FragmentContainerView for the NavHostFragment to the activity's xml Access the NavHost's corresponding NavController using the findNavController function Edit the project's gradle file to enable SafeArgs Use NavDirections object when navigating Add arguments to destinations in the navigation graph Attach the argument to the navigation within the fragment	Navigation slides - Code-along demo app throughout slides Navigation programming lab Navigation quiz Documentation Day programming activity 1 Documentation Day programming activity 2	Q&A during slides Code-along demo app Navigation programming lab Navigation quiz Documentation Day programming activity 1 Documentation Day programming activity 2

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
NAVIGATION COMPONENT (continued)	What is the risk of never popping an activity/fragment off of the backstack? What is the risk of creating a new instance of the previous fragment when we navigate back to it? In which destination do you create the FragmentResult Listener? In which destination do you call setFragment result?	Fragment Result API - Fragment ResultListener - setFragmentResult Listener _____ (*Documentation Day 1)* Options Menu (*Documentation Day 2)* MediaPlayer	Retrieve the argument from the originating destination from within the receiving destination Use navigateUp to navigate back to the originating destination Edit the project's gradle file to enable the Fragment Result API Navigate back to the originating destination while passing information back, using the Fragment Result API Create an anonymous FragmentResultListener by calling setFragmentResultListener within the originating destination Set the result to be passed back within the receiving fragment by calling setFragmentResult Get the result within the originating fragment _____ *(Documentation Day 1)* Add an options menu within the app bar (*Documentation Day 2)* Add sound to an app that the user can start, pause, rewind, and fast forward		

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
RECYCLER VIEW	What type of class should we define for the model? What attribute must the RecyclerView include in order to be visible on the screen? Why is it important to make the height of the list item layout "wrap_content" instead of "match_parent"? What is the main job of the ViewHolder? What is the purpose of the adapter? What three functions must you override to implement a RecyclerView adapter?	The concept behind RecyclerView Creating the model for a RecyclerView - Data classes How to add the RecyclerView dependency How to create a RecyclerView in layout LayoutManagers - LinearLayout Manager How to create a layout file to describe how a single row should look - Height attribute ViewHolders - RecyclerView. ViewHolder - Constructor argument - Creating a ViewHolder class	Define an object to model the data being displayed in the list Add the RecyclerView dependency to the project's gradle file Create the RecyclerView in the activity/fragment layout file Set the layout manager of the RecyclerView Create the layout file for a single list item Create a ViewHolder class to manage the list item's layout file Create a bind function within the ViewHolder to connect model data with the list item layout's views Create an Adapter class Override the onItemCount function and return the list size Override the onCreateViewHolder function to inflate the list item's layout and use it to create a ViewHolder instance Override the onBindViewHolder function to bind the model object at the current position to a view	RecyclerView slides - Code-along demo app throughout slides RecyclerView programming lab RecyclerView quiz Documentation Day programming activity	Q&A during slides Code-along demo app RecyclerView programming lab RecyclerView quiz Documentation Day programming activity

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
RECYCLER VIEW (continued)	Which method gets called more often - onCreateViewHolder or onBindViewHolder? Why? Where do you create the list of data the list will hold? Where do you create a click listener to respond to list presses?	Adapters - RecyclerView.Adapter - Constructor argument - getItemCount - onCreateViewHolder - onBindViewHolder - onCreateViewHolder vs onBindViewHolder - Creating an instance of the adapter Connecting the adapter and RecyclerView Responding to the user pressing a row within the RecyclerView - ViewHolder init block _____ (*Documentation Day)* Taking a photo	Declare a list of model objects and use it to create an adapter instance Connect the adapter to the RecyclerView by setting the RecyclerView's adapter property Set a click listener on the root view within the view holder and take action when a list item is pressed _____ (*Documentation Day)* Use an implicit intent to launch the camera app and take a photo		

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
VIEWMODEL AND LIVEDATA	What are the advantages of good app architecture? What are the main principles of good app architecture? What classes fall under the category of UI Controller? How many ViewModel instances can there be per UI Controller? Can the ViewModel hold a reference to the UI Controller? Can the UI Controller hold a reference to the ViewModel? What's the advantage of using property delegation?	Android app architecture - UI controller - ViewModel Creating a ViewModel - Kotlin property delegation - Shared ViewModel Properties in UI Controller vs ViewModel - Kotlin backing property ViewModel lifecycle Functions in UI Controller vs ViewModel LiveData - LiveData vs MutableLiveData Observing properties ——————— (*Documentation Day)* AlertDialog	Create a class that inherits from Android's ViewModel class Create a ViewModel property within the activity/fragment Initialize the ViewModel property using property delegation Determine if a property should be declared in the UI Controller or ViewModel Declare a public read-only backing property for an editable private property Access ViewModel properties from within the Activity/Fragment Determine if a function should be declared in the UI Controller or ViewModel Access ViewModel functions from within the Activity/Fragment Declare a property whose type is LiveData Declare a property whose type is MutableLiveData Access the LiveData's value property Change the value of a MutableLiveData property	ViewModel and LivaData slides - Code-along demo app throughout slides ViewModel and LiveData programming lab ViewModel and LiveData quiz Documentation Day programming activity	Q&A during slides Code-along demo app ViewModel and LiveData programming lab ViewModel and LiveData quiz Documentation Day programming activity

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
VIEWMODEL AND LIVEDATA (continued)	What criteria should you use to determine if a property/function belongs in the UI Controller or View Model? Why do we use Kotlin backing properties? Does the ViewModel instance persist lifecycle changes? What's the advantage of using LiveData?		Observe a ViewModel property from within the Activity/Fragment Use the parameter within the observer's callback function ——————————— (*Documentation Day)* Use an AlertDialog to display a pop up alert with buttons		
NETWORKING	What is a protocol? What is a web service? What's a client vs a server? What's the format of the HTTP response? What are the four HTTP actions?	Networking Basics - Networking terms - Client and server - HTTP requests - HTTP actions - APIs - URLs - Queries Permissions - Internet permission Introduction to Retrofit	Define a Retrofit service as an interface Define a function within the interface for each HTTP request Use annotations to specify the HTTP action of a function within the interface Given an API, identify the base URL for a request Given an API, identify the endpoint for a request	Networking Part 1 slides - Code-along demo app throughout slides Networking Part 1 programming lab Networking Part 1 quiz ————————	Q&A during slides Code-along demo app in Networking Part 1 Networking Part 1 programming lab Networking Part 1 quiz ————————

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
NETWORKING (continued)	What is an endpoint? What are the four parts of an URL? What symbol begins a query? What separates a query's key and value? What's the advantage of using Retrofit? What type of response do you get when using a Scalar converter? ——————— What's the advantage of using Moshi instead of ScalarConverter? Why do we need to specify for Moshi what format (e.g. JSON) the response is in?	Defining the Retrofit API service (i.e. interface) - Creating a function for each HTTP request - Call object - Defining the query Building a Retrofit object - Base URL - Response converter - Scalar converter - Converter factory Creating an API service instance Executing the web request - Enqueue - onResponse and onFailure ——————— Moshi - Creating a moshi instance - Linking to retrofit Introduction to JSON - JSON components - JSON objects - JSON arrays - Extracting specific data	Given an API, identify the query to specify parameters for a request Create a Retrofit object using the build function Specify the base URL for a Retrofit object Specify the response converter for a Retrofit object Create a companion object attached to an interface Use the enqueue function to start a HTTP request Override the onFailure and onResponse methods to handle the response ——————— Create a Moshi object using the build function Specify the format of JSON by adding the KotlinJsonAdapterFactory Identify if the value for a JSON key is a primitive or object type Identify if the value for a JSON key is a JSON object or JSON array	Networking Part 2 slides - Code-along demo app throughout slides Networking Part 2 programming lab Networking Part 2 quiz Documentation Day programming activity	Code-along demo app in Networking Part 2 Networking Part 2 programming lab Networking Part 2 quiz Documentation Day programming activity

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
NETWORKING (continued)	What symbols denote a JSON object? What symbols denote a JSON array? What types of JSON values should you make custom Kotlin types for? How do you represent a JSON array in Kotlin? How do you handle the possibility that the response is null?	Preparing to convert the JSON to Kotlin objects with Moshi - Identifying the keys for each node - Defining Kotlin types for JSON components Parsing the JSON with Moshi - Setting up the response - Creating the response LiveData - Changing the response type - Parsing the response - Observing the response Loading a web page	Trace the path from the root response to the desired information by specifying the keys and their types along the path Use annotations to map custom Kotlin objects to JSON keys Create a custom Kotlin type for the root response Represent the hierarchical structure of the JSON by specifying properties for the custom Kotlin objects Create a LiveData and MutableLiveData object to store the response Observe the response object Convert a URL to a URI and use an implicit intent to load a webpage		
FINAL PROJECT	N/A			Final Project description document Final Project research Final Project idea form	Final Project progress journal document Final Project learning summary document Final Project resources document

	Essential Questions	Content	Skills	Activities	Assessment / Evidence of Learning
FINAL PROJECT (continued)				Final Project progress journal document Final Project learning summary document Final Project resources document	