CODE TRAIN

Note for our AP CS Principles Teachers

In September 2023, the College Board announced significant changes to the way the Create Task and the Exam are structured for this year and moving forward. As these changes were announced after the school year has started, our lesson plans docs, including this one, may not reflect the changes. Those docs will be updated throughout the year as our own teaching schedules allow.

On October 27, 2023, CodeTrain presented to our teacher base describing these changes and updates. That presentation can be found here. Please contact info@codetra.in if you would like more information or get on our mailing list for future PD.

NEW: In February 2024, CodeTrain produced 3 new resources for the updated Create Task and Exam.

- Create Task Project Options
- Create Task Submission Requirements
- CSP Free Response Sample Exam Questions

Those resources can all be found in the Create Task folder which can also be found in our 2023 Code Train Shared Activities folder.

NEW: In March 2024, CodeTrain produced 2 companion videos for the CSP Create Task Updates. Those videos are available free but we must track who watches those videos for funding purposes. To request access, please email:

To: Info@codetra.in
Subject: CSP Create Task Update Videos

NEW: In April 2024, The Lesson Plan packages, including this doc, will be updated to reflect the new resources.

Updated Java Activities Available in our CSA Package

Please note that this is our oldest lesson plan package. In the Summer of 2023, we

released our CodeTrain CSA Resources package which also focuses on Java. While many of the activities are similar, the CSA package does contain many new lessons and assignments. To view the CSA package, please return to our resources page.

CodeTrain Package 1

- - -

Two Year Computer Science Program Using Processing & Java

- - -

Year B

Part B of a Two Year program that fulfills the requirements of Advanced Placement CS Principles

Last Update March 2024

How to Use this Resource

This package is designed to be supported by our CodeTra.in professional development sessions. For ease of use and to get the most value from these resources, please contact us for our low or no cost in-person and online PD options.

This document contains lessons and activities designed for 75 classes of approximately 80 minutes. It covers the second year of a 2 year program and can be used in conjunction with the Year 1 Lesson Plan to fulfill the requirements of the AP CS Principles Course.

It is a resource created for teachers to help them build a successful and exciting Computer Science program. It is not intended to be used directly by students.

The AP CS Principles Course and Exam Description can be found here. In order to ensure all of the AP outcomes are met, many are listed directly in the lessons and are in 9pt Calibri font. At the time this document was produced, these outcomes were still in draft form so there may be some slight differences in the wording once the official document is published.

See the front matter of the Year 1 Lesson Plan document for more information about the other provided resources.

Year Plan and Upcoming Changes

The course is split into 2 halves

1. Units 1-4 cover new programming concepts
2. Units 0, 4-6 cover theory and the 2 AP Assessments

Units 0-3 are currently undergoing an evolution process in my actual classroom. In these lessons, we switch to Java immediately. In my actual classroom, I stay in our first language (Processing or P5.js) for most of the course, with a single 'Intro to Java' unit at the end to prepare students for a transition into a third year which gives students the opportunity to take AP CSA. Due to the disruptions of 2020/21, I switched to using P5.JS in my first year and our newer One Year program covers the entire CSP curriculum in P5.js. However, I still use a 2 year CSP program in my classes.

Table of Contents

Unit Zero - Introduction to Algorithms

This content is described as Unit Zero as it is not a set of formal lessons. Instead, it is a set of interesting videos and discussions that you can use to intermingle throughout the coding units of the course.

Some videos cover specific AP content. Others just discuss the power and impact of algorithms on modern life.

They are grouped in 2 sets mirroring the Big Ideas of Algorithms and Global Impact. The videos can be intermingled and should be peppered in throughout the year in order to cover all the course content.

Unit Zero Video Set 1
Algorithms and Their Impact

0.1. Introduction to Algorithms - Moving from Coders to Developers

- Now that you know the basics for programming, it is time to learn how to program for more complex tasks. Everything we do this year, is designed to move you in that direction.
- Big Bang Theory Algorithms (warn about spelling errors) https://www.youtube.com/watch?v=lbBIKGpQqcY
- http://www.csfieldguide.org.nz/en/chapters/algorithms.html
    - Does the binary search algorithm remind you of any assignment we did last year? Is it not similar to how to solve the Number Guess game?
-

0.2 Global Impact - "The Secret Rules of Modern Living Algorithms" - Documentary (58 minutes)

- This excellent hour long documentary is **Highly Recommended** and therefore also appears with more detail in Lessons 1&2 of this document.
-

0.3 Intro to Algorithms: Crash Course Computer Science #13 (11:43)

0.4 Unintended Impact of Algorithms: How Algorithms Shape our World. (15 minutes)

- One of the Big Ideas is Global Impact, this video focuses on the impact of technology that they might not have considered.
- The video focuses on the stock market. Please ask them *What does it mean to trade stocks? How do you make money trading stocks?* Then just take a couple of minutes ensuring everyone knows the basics.
- Tell them to *Put Your Thinking Caps On* and then start the video.

0.5 Algorithms and what they can not solve

- This video is very challenging and may only be approachable by more academic students. It appears with more detail in Lessons 25 of this document.
- P vs NP - https://youtu.be/YX40hbAHx3s (11 minutes)

Unit Zero Video Set 2
Global Impact & Emerging Technology Videos

Global Impact was discussed throughout Year 1. Below are a new set of suggested videos to show at various points throughout the course to continue these discussions.

0.6 Global Impact Video - Big Data Revolution Documentary
- This excellent hour long documentary is **Highly Recommended** and therefore also appears with more detail in Lessons 21-23 of this document.

0.7 Global Impact - Algorithmic Bias
- Search Grandma on Google and what do you see? Probably the stereotypical white granny. Add the filter 'cute' and the results are even worse
- Why are the search results disproportionately white?
- How I'm fighting bias in Algorithms (8:43) https://www.ted.com/talks/joy_buolamwini_how_i_m_fighting_bias_in_algorithms
- When white males start a project, their data set is their friends and acquaintances. Their data sets are often the problem, not their own biases. Algorithms therefore adopt the unconscious skews and biases of their developers, unless developers are hyper conscious of this potential problem.
- Problems of bias are often created by the type or source of data being collected. Bias is not eliminated by simply collecting more data.
- Computing innovations can reflect existing human biases because of biases written into the algorithms or biases in the data used by the innovation.
- Programmers should take action to reduce bias in algorithms used for computing innovations as a way of combating existing human biases.
- Biases can be embedded at all levels of software development.

0.8 Global Impact Video - **How Smart is today's Artificial Intelligence (7:52)**
- Good explanation of machine learning

0.9 Global Impact Video - Quantum Computer - How Does it Work (6:46)
- Moore's Law has proven true for 5 decades
- It has given the ability for industries to plan for the future by banking on exponential growth
- However, it is nearing its end.
- Quantum Computers have been discussed as a successor. This video explains the basics of how they work and why they will NOT replace our current computers.

0.10 Global Impact Video - Block chain
- Youtube: Understanding Block chain (2:30)

- Youtube: 19 Industries that will be disrupted by block chain (9:30 but probably only about 6:00 is sustainable)

0.11 Global Impact Video - AlphaGo Documentary

- Available on Netflix?
- This surprisingly entrancing full length documentary leads to lots of philosophical questions on the future of A.I.

0.12 Global Impact Video -Ray Kurzweil Get Ready for Hybrid Thinking (9:49)

- Show the trailer for Transcendent Man
- Discuss how he has invented the scanner, reading machines, Kurzweil Keyboard, etc.
- Ray Kurzweil says you should invent not with the technologies that are available today, but with the technologies that will be available in the future.

Year 2 Unit 1 - Intro to Java - 8 Classes

This unit introduces students to programming in Java. I use the Netbeans IDE but there are lots of others. If you use another IDE, you will need to import my Java files appropriately.

I use the Scanner class for user input. You will see in my lessons that I use 2 instances of the Scanner class called numscan and wordscan for inputting numbers and Strings respectively. I do this to avoid buffering issues when teaching beginners.

Before continuing, please read the notes at the beginning of this document regarding my evolving transition to Processing in Year 2.

Remember to insert the Unit Zero videos at various points throughout the lessons. In some cases, these can be used to supplement sort lessons. In other cases, it may extend the suggested length of the unit.

Unit Quick Links
Full Activities Folder
Direct Link to Teaching Examples folder (must be downloaded)
Student example assignments - Available to teachers who attend our PD sessions

Year 2, Day # 1	Class #1 of Year 2 Unit 1

1.1 Welcome and introductions

- This course will begin to move you from just being programmers, towards being Computing Scientists. That means you will be regularly challenged to make your code more efficient. You will also be asked to extend your projects with extra algorithms that will challenge your thinking.
- Go over course syllabus and marking schemes
- Talk extensively about Java requiring a step back. We will only write TEXT based programs for the first unit, then will transition back into Processing.

1.2 Algorithms Video - "The Secret Rules of Modern Living Algorithms" - Documentary (58 minutes total)

- Show half of this video today, and half in lesson 2
- This excellent documentary is sometimes available on Netflix but otherwise available at on Youtube.
- It has a particularly nice explanation of how the algorithm that changed the world works: Google's Page Rank
- Discuss/review the following AP CSP Essential Knowledge Statements:
    - Algorithms can be written in different ways and still accomplish the same tasks.
    - Algorithms that appear similar can yield different side effects or results.
    - Algorithms can be created from an idea, by combining existing algorithms, or by modifying existing algorithms.
    - Knowledge of existing algorithms can help in constructing new ones.
    - Using existing correct algorithms as building blocks for constructing another algorithm has benefits such as reducing development time, reducing testing, and simplifying the identification of errors.
    - An algorithm is a finite set of instructions that accomplish a specific task.
    - Beyond visual and textual programming languages, algorithms can be expressed in a variety of ways, such as natural language, diagrams, and pseudocode.
    - Algorithms executed by programs are implemented using programming languages.
    - Every algorithm can be constructed using combinations of sequencing, selection, and iteration.

1.3 Quick review in Processing to review the concept of an Algorithm - Factorials (code on next page)

- Use a println statement to do a factorial of 10 (multiple all the numbers up to 10)
- Then do it in a FOR loop
- Then do it in a function
- Then do it in a function that takes in a parameter
- Find out the factorial size that loops beyond the size of an int (17) and discuss the wrapping of the variable and briefly mention how it relates to 2s Complement binary

- wrapping of the variable and briefly mention how it relates to 2s Complement binary

1.4 Introduction to Netbeans (or your choice of IDE and Java)

- Discuss the use of the Netbeans project folder. We will be using it
- Explain the use of Projects and packages

1.5 Java Lesson 1 - Recreate the Factorial Function in Java

- See Ex1_Factorial.java in the Examples Folder
- They will see it gets a bit complicated looking so we will take a step back for a couple of days, focusing on getting to know the differences between Java and Processing

1.6 Java Lesson 2 - Introducing the Scanner Class

- What is a class? What does an import statement do.
- *Scanner used to ask the user what number to factorialize*

Processing Example 1 - Factorial

```
//Factorial
//Multiply the numbers up to and including the number
//int number = 10;
void setup() {
  factorial(17);
  factorial(10);
  factorial(6);
}
void draw() {
}
void factorial(  int number     ) {
  int total = 1;
  for (int i=1; i<=number; i++) {
    total *= i;    //total = total * i;
  }
  println("Total is "  + total);
}
```

Year 2, Day #2	Class #2 of Year 2 Unit 1

2.1 Scanner Lesson 2

- Ex2_Grades.java
- Walk them through the differences between numscan.nextInt() and wordscan.nextLine()
- Discuss common mistakes including using the wrong scanner
- Introduces String.equals() and String.equalsIgnoreCase() for String If statements

2.2 Java Lesson 3 - Inputting Numbers and While Loops with the Number Guess Game (see below for code)

- Discuss/review the following concepts:
    - Iteration is a repeating portion of an algorithm. Iteration repeats a specified number of times or until a given condition is met.
    - Iteration statements change the sequential flow of control by repeating a set of statements zero or more times, until a stopping condition is met.
- Ex3_NumberGuessGames
- Last year, they completed this in Processing as an assignment.
- Have them work on it as a review activity with the following help:
    - Give them the random number generator
    - Show them while(true) { break; }

2.3 Algorithms Video Part 2

- Conclude the video from last lesson.
- Reinforce the following concepts:
    - An algorithm is a finite set of instructions that accomplish a specific task.
    - Beyond visual and textual programming languages, algorithms can be expressed in a variety of ways, such as natural language, diagrams, and pseudocode.
    - Algorithms executed by programs are implemented using programming languages.
    - Every algorithm can be constructed using combinations of sequencing, selection, and iteration.

2.4 AP Classroom Practice Questions, Topic Questions and Progress Checks

- AP Classroom has a number of Practice Questions aligned to the content. We recommend using applicable practice questions at the end of each programming unit.
- Teachers are reminded that AP Classroom Progress Checks are to be used for Formative assessments only.
- AP Classroom let's you filter questions by Big Idea for CSP. Remember that they are not specific to any language but serve as a practice for the CSP exam. If you also have students completing CSA, you can access specific Java questions through the CSA Classroom.

Year 2, Day #3	Class #3 of Year 2 Unit 1

3.1 Algorithms Can be Expressed in a number of ways

- Algorithms are sequences of statements that accomplish a specific task.
- Therefore they can be expressed without programming languages
- Pseudocode and flowcharting are two examples of ways to express an algorithm without the use of code
- Demonstrate using a flowchart to express the Number Guess game program's algorithm.
- Good sample images can be found here: https://www.smartdraw.com/flowchart/flowchart-symbols.htm

3.2 Java Lesson 4 - More String Methods

- See code on next page
- Use a UserName and Password String
    - Ask if they are usually case sensitive
    - Start with usernames and use .equalsIgoreCase () first
    - Then add .toLowerCase() and discuss how you may want to store the username longterm as lower case letters
    - Then add the && sign for checking both.
    - Use contains
    - Use startswith

- Copy that code and then add a while loop that will not allow the user through until the username and passwords match
- Discuss using an array to contain a list of usernames and passwords but we will do this later.

3.3 Java Assignment 1 - So You Think You Can Rap

- The first assignment is not exceptionally challenging but it ensures they understand the basics of Java input and output and reviews the basic data types from Year 1.
- See separate doc in Resources Folder

Year 2, Day #4	Class #4 of Year 2 Unit 1

4.1 Recommended - show a Global Impact Video from the Theory Unit or Unit Zero

4.2 Quick addition to yesterday's example - pull off the year and discuss parseInt

```
year = date.substring(date.length()-4);
System.out.println("Year: " + year);
 int newyear = Integer.parseInt(year);
```

- Also demonstrate the date or calendar class. Sample code below from https://examples.javacodegeeks.com/java-basics/java-calendar-example/
- Review the use of Nested Conditionals (IF statements)
- Nested conditional statements, or "else if" statements, consist of conditional statements within conditional statements.
- If the Boolean condition of the initial conditional statement evaluates to false, then the Boolean condition of the nested conditional statement is evaluated.

4.3 Java Assignment 2a - How Cool are You Part 1 - Flow Chart

- See separate doc in Resources Folder

4.4 Java Assignment 2b - How Cool are You Part 2 - Flow Chart

See separate doc in Resources Folder

Year 2, Day #5	Class #5 of Year 2 Unit 1

5.1 Java Lesson - For Loops:

- Ex5_ForLoops.java
- Count up
- Count down
- Squares of i
- Find all the numbers that are evenly square rooted between 1 and x. (introduce MOD) - See next page
- Intro to Arrays

5.2 Finish Java Assignment 2 - How Cool are You

5.3 Start Java Assignment 3 - Ninety-Nine Bottles

- See separate doc in Resources Folder

Year 2, Day #6	Class #6 of Year 2 Unit 1

6.1 Discuss the upcoming No Help Assignment/Test

- *Ask if anyone would like a paper copy or if a PDF is fine.*
- Dividers will be up
- You may use old code
- You may ask for help with syntax and I will let you know if that is a reasonable question. You will likely not get help with logic or ‘how do I…’ questions

6.2 Shopping Cart Example - While Loops and Switch Statement

- Ex5_ShoppingCart.java
- Discuss that this leads to a marked assignment.
- Complete the switch component first (ask students to run and debug), then add the loop part of the program
- After choosing an item, ask how many they want.
- Using a while loop, ask them if they want to keep shopping or choose from a list of items.
- Update a running total and ask again, redisplaying the menu of choices.
- Once they set keepshopping variable to false, display the total.

6.3 Java Assignments Catch Up Time

Year 2, Day #7	Class #7 of Year 2 Unit 1

7.1 Recommended - show a Global Impact Video from the Theory Unit

7.2 Java Assignment 4 - Shopping Cart

- See doc in Resources Folder
- Requires yesterday's example as a starting point

7.3 Java Assignment 5 - Text Adventure

- See doc in Resources Folder

Year 2, Day #8	Class #8 of Year 2 Unit 1

8.1 Java Intro Unit No Help Assignment/Test

- See doc in resources folder
- Note that this is quite similar to the How Cool assignment. It focuses on IF statements and while loops and does not test the For Loop or Switch statement.

Unit 2 - Intro to Functions - 12 Classes

As stated in the introduction, this unit is entirely written in Java for now. During the course of the 2019/20 school year, Processing materials will become available.

The general philosophy behind the progression of the unit is to isolate the various components to ensure mastery before combining them and increasing the complexity. Those components are:

- Simple void methods - mostly a review of Year 1
- Passing Parameters
- Returning Values
- Parameters and Return Values Combined

Remember to insert the Unit Zero videos at various points throughout the lessons. In some cases, these can be used to supplement sort lessons. In other cases, it may extend the suggested length of the unit.

Unit Quick Links
Full Activities Folder
Direct Link to Teaching Examples folder (must be downloaded)
Student example assignments - Available to teachers who attend our PD sessions

Year 2, Day #9	Class #1 of Year 2 Unit 2

9.1 Review Void Methods with out Parameters

- Void methods were introduced in year 1. This review is just to get students used to the Java syntax.
- Begin by discussing the terms Parameters (aka arguments) and return values and how they are written in Java:

```
/*
 public static returnType methodName ( parameters ){
    returntype - a value you send back int, double, string, boolean, void etc.
    parameters - values you send IN to the method
 }
 */
```

- Discuss the synonyms for a Function: procedure, subroutine, etc. Java tends to use the word METHOD. Some hardcore Computer Scienctists may argue that there are technical differences between these terms. AP CS Principles uses the word PROCEDURE
- A procedure is a named group of programming instructions that may have parameters and return values.
- Procedures are reusable abstraction of a task
- A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has executed, flow of control is returned to the point immediately following where the procedure was called.
- Parameters are input variables of a procedure. Arguments specify the values of the parameters when a procedure is called.
- Ex1_VoidMethods.java is one way you can cover these concepts

9.2 Optional Assignment: Methods 0 - Void Methods Practice

- See doc in resources folder

Year 2, Day #10	Class #2 of Year 2 Unit 2

10.1 Discuss the advantages of Modularity and Procedural Abstraction including these AP CSP Essential Knowledge Statements

- The subdivision of a computer program into separate subprograms is called modularity.
- Procedural Abstraction is one common type of abstraction. It provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it.
- Procedural abstraction allows a solution to a large problem to be based on the solution of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems.
- A procedural abstraction may extract shared features to generalize functionality instead of duplicating code. This allows for program code reuse, which helps manage complexity.
- Using procedural abstraction helps improve code readability.
- Using procedural abstraction in a program allows programmers to change the internals of the procedure (to make it faster, more efficient, use less storage, etc.) without needing to notify users of the change as long as what the procedure does is preserved.
- Procedural Abstraction is related to the larger CS concept of Encapsulation (which is not a CS Principles term)

10.1a Class discussion on how Procedural Abstraction manages complexity in a program

- Have students discuss in partners when we used Functions last year
- If they need prompting, remind them about the scene functions and more complex projects like Hunger, Rock and Tic.
- How would the code be more complicated without the use of functions?

10.2 Introduce void Methods with Parameters

- Using parameters allows procedures to be generalized, enabling the procedures to be reused with a range of input values or arguments.
- See Ex3_MathParameters.java
- printArea(int diameter)
- printCircumference(int diameter)
- printArea(int length, int width) - discuss the term overloading
- printVolume(int diameter, int height, String units)
    - *the cylinder's volume is ___, cubic centimeters*
    - then add an if statement if height ==0, print
- Discuss the following concepts:
    - Parameters are input variables of a procedure. Arguments specify the values of the parameters when a procedure is called.

10.3 Discuss the Java terms public and static

- Note: the terms *public* and *static* are not part of the AP CS Principles outcomes but are important when discussing Object Oriented Programming in Java.
- Both terms will become more clear once we start using Object Oriented Programming

- PUBLIC is a type of Accessibility Indicator. It means that this method (or variable) is useable by any other Java class in the project. Other Accessibility Indicators (such as PRIVATE or PROTECTED) limit the useability by other classes.
- STATIC is tough to explain until students get an understanding of creating Instances of OOP classes. STATIC is an indicator that this thing does not require an instance to be created before using it.

10.4 Parameters Example - Banking without Return Values

- See Ex4_Banking_WithoutReturnValues.java
- This example is designed to reinforce parameter passing BUT has the secondary purpose of showing the limitations of methods when return values are not used.
- Students will quickly see that the BALANCE variable is not updated after each transaction. After the next assignment, we will fix this issue by having our function RETURN the updated balance.
- The example as written also creates Scanner classes as GLOBAL variables. Scanners are the only Global variables allowed during this unit.

Year 2, Day #11	Class #3 of Year 2 Unit 2

11.1 Using Libraries of Procedures and APIs

- The Java API is often called ‘javadocs’ by programmers
    - Application program interfaces (APIs) are specifications for how the procedures in a library behave and can be used.
    - Documentation for an API/library is necessary in understanding the behavior(s) provided by the API/library and how to use them.
- Tour the Java API, pointing out all the libraries along the left side that are available to programmers
    - These are the standard java libraries. Non-standard libraries are those built by others that can be imported into your program. Often these libraries are open source but sometimes are sold by the developer.
    - Examine the Scanner and the String classes and pay close attention to the Methods included in each of these classes
    - A software library contains procedures that may be used in creating new programs.
    - Existing code segments can come from internal or external sources, such as libraries previously written code.
    - The use of libraries simplifies the task of creating complex programs.

11.2 Void Methods Review - Quick Challenge:

- The Rap challenge - take in 2 lines and award a prize if it ‘rhymes’. For us, rhymes mean the last 3 letters match.

11.3 Methods Assignment 1 - The Restaurant

- This assignment focuses students on Parameter passing. Once again, it will show the limitations of methods that do not use return values.

Year 2, Day #12	Class #4 of Year 2 Unit 2

12.1 Void Methods Review - Quick Challenge:

- RoundMe- Takes in 2 parameters, a double and an int. It rounds the double to the correct number of decimal places and just prints the correct result.

12.2 Methods Assignment 2 - Home Reno

- This assignment continues to focus students on Parameter passing.
- It is easily adaptable should you want students to start using simple Return Values

12.3 Discuss Procedural Abstraction

- Remind them of the following
    - AAP-3.B.1: One common type of abstraction is procedural abstraction, which provides a name for a process and allows a procedure to be used only knowing what it does, not how it does it
    - AAP-3.B.2: Procedural abstraction allows a solution to a large problem to be based on the solution of smaller subproblems. This is accomplished by creating procedures to solve each of the subproblems.
    - AAP-3.B.4: A procedural abstraction may extract shared features to generalize functionality instead of duplicating code. This allows for program code reuse, which helps manage complexity.
    - AAP-3.B.6: Using procedural abstraction helps improve code readability.
    - AAP-3.B.7: Using procedural abstraction in a program allows programmers to change the internals of the procedure (to make it faster, more efficient, use less storage, etc.) without needing to notify users of the change as long as what the procedure does is
- In partners, have students discuss the following questions by making reference to their Restaurant and Home Reno Assignments
    - What does procedural abstraction mean?
    - Do you think your methods help manage complexity? Improve code readability?

Year 2, Day #13	Class #5 of Year 2 Unit 2

13.1 Method Call vs Declaration

- At this point, students start to get confused about the difference between Calling a Method and Declaring a Method.
- The activity below has them practice these concepts.
- Students write the corresponding method call or method declaration line (not the contents of the method itself). Give freedom of variables names and types where appropriate (doubles may be accepted for ints, etc).
- We do so one at a time and discuss the answers immediately afterwards. A similar activity will occur in a couple of classes time once they have practiced return values.

What would the call from Main have to look like to call the following methods

Public static void placeOnMap(double lat, double long){ }
Public static void printInitials(String fn, String ln){ }
Public static void findCharacter(String word, int charNum){ }

What would the top of method declaration have to look like if the following calls came from main

callHome("555-2454")
addTax(299.90)
breakRecord(42.5, 46)
buyForCouprie("Porsche", 2)

13.2 Catch up time on Assignments

13.3 Introduction to Return Values - Bank Account Redeux

- See Ex4_Banking_WithReturnValues.java
- Open the original Banking example and then demonstrate *Refactor… Copy* and discuss the other uses of Refactoring
- Update the original example to return the balance

- Once this basic update is working, challenge the students to work in the TWO bank accounts so that the user can choose an account and the proper balance is updated.

Year 2, Day #14	Class #6 of Year 2 Unit 2

14.1 Return Values Continued

- Reiterate the sending in of parameters and the sending out of return values
- The RETURN statement may appear at any point inside the procedure and causes an immediate return from the procedure back to the calling statement.
- Remind them of the updates we made to Banking
- Discuss how return values need to be used by the calling method either by:
    - Saving the result into a variable to be used later
    - Or used immediately such as in a println or in a boolean statement

- Discuss how it would have helped on the Restaurant assignment
- Discussion - What do you think the return value is of each of the following functions:
    - .contains booolean
    - .toLowerCase String
    - .substring String
    - .parseInt int
    - random double

14.2 Return Values Example

- See Ex5_ReturnValues.java
- This unrelated collection of methods is designed to show how Parameters and Return values are used for many different applications
- String colorConverter(int r, int g, int b)
- Boolean isPassing (int mark1, int mark2, int mark3)
    - Discuss that Boolean Expressions in return statements can be used in place of IF Statements
    - In other words return x>5 will return the true or false
    - This is the equivalent of if (x>5) {return true} else { return false;}
- Double addTax(double total, String province)

14.3 Return Methods - Quick Challenge:

- String pigLatin(String english)- Takes in an English word or set of words and returns pig latin

14.4 Got Time for a Video?

- Choose from an Algorithms or Global Impact (Theory) video

Year 2, Day #15	Class #7 of Year 2 Unit 2

15.1 Static Class - Input Example

- This is a first taste of Object Oriented Programming
- Discuss the idea of using classes to organize your code
- This INPUT class uses pop up windows are an alternative to Scanners and Printlns
- But it can be annoying to go back and forth. We will therefore show you how to use them but add a running Printout to the console to help.

15.2 Introduce the Final 10% assignment for the Unit

- Introduce Javadocs in preparation for the Final 10% assignment
- This assignment gives students open ended tasks, thus encouraging them to find creative solutions. Give them ideas on how to plan and design their program before beginning their coding process.
- This project focused on developing the following AP CSP Enduring Understandings, amongst others:
    - CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/ feedback cycles, and that leaves ample room for experimentation and risk-taking.
    - AAP-3 - Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

15.3 Return Values Assignment Simple Calculator

- See doc in Resources folder
- Be sure to walk them through the requirements
    - must return the value to main and print it in main
    - no parameters
- Note that Simple Calculator is still a big leap as they are not sure how to set things up.
- Do a full walk through of setting up all 4 method declarations, the scanners etc.
- Also set up the ANSWER variable and have it printed after the 4 IF STATEMENTS
- They are then NOT allowed to change the method declaration. *Are you allowed to change the method declaration? NO!*

Year 2, Day #16	Class #8 of Year 2 Unit 2

16.1 Return Values Example: Card Game

- Uses a Games utility class (Games.java) plus a main class (Ex6_CardGame.java)
- This is a major example, best split over a couple of days. It is needed for a future assignment.
- Create a Games class for all types of games
    - getRandom(int highest)
    - getRandom(int lowest, int highest)
    - getSuit()
    - printCard(int cardNum, String suit) - converts 11 to jack, etc.

16.2 Introduce Assignment 5 Olympics

- The activity is not much more difficult BUT I give you a little less OBVIOUS information about how the methods should be declared

16.3 Work time on the following assignments:

- Simple Calculator
- Olympics
- Final 10%

16.4 Quick Review Questions for upcoming exam:

- What are 2 synonyms for the term *method*?
- Can a method return more than one thing?
- Why is limiting the use of global variables important?
- What is procedural abstraction?
- How does procedural abstraction help manage code complexity?

Year 2, Day #17	Class #9 of Year 2 Unit 2

17.1 Finish Card Game Example

17.2 Work time on all assignments

17.3 Quick Review Questions for upcoming exam:

- We can have more than one method with the same name under what circumstances?
- What is this concept called?

Year 2, Day #18	Class #10 of Year 2 Unit 2

18.1 Call vs declaration practice - Return Values Version

- This is a follow up activity. See the instructions on Class #5 of the unit.

```
Methods Call and Declarations - Return Values
main () {
        String todaysDate;
        todaysDate = getDate();

        System.out.println("The Raptors points are" + getWins( ) * 2);

        if ( checkAbsences("Suzie")  > 6 ) {
                System.out.println("Call home");
        }

        if (  isPassing("Suzie", 12)   ){
                System.out.println("Begin next credit");
        }

        myaverage = _____;
}//end main

public static double getAverage (int assign1, int assign2, int assign3){ }
```

18.2 Tracing Code Activity 3

- Walk students through the Exam syntax and let them test their understanding with Khan Academy.

18.3 Introduce Assignment 6 Life Expectancy/Casino

- The Casino version builds off the in-class Card Game example.

Year 2, Day #19	Class #11 of Year 2 Unit 2

The Unit concludes with a two-part exam. The first is a No Help Assignment similar to those used in earlier units. The second is the paper exam.

19.1 No Help Assignment/Exam - Triangle Tester

- See doc in the Resources folder
- The activity is not exceptionally difficult but does ensure they understand the interplay between methods, including boolean methods.

- Walk them through the requirements before posting
- -10% for help with old code
- Discuss the flow but tell them they can get good part marks if they can make each method work but not in the manner discussed in Main. (If they come up with their own way to test).
- It is their responsibility to identify the inputs and outputs necessary to fully test the code.

19.2 Other End of Unit Activities:AP Classroom Practice Questions, Topic Questions and Progress Checks

- AP Classroom has a number of Practice Questions aligned to the content. We recommend using applicable practice questions at the end of each programming unit.
- Take a few moments to go over the Topic Questions for 3.12 Calling Procedures and 3.1 Developing Procedures. There were 30+ sample multiple choice questions available at the time this document was produced.
- Teachers are reminded that AP Classroom Progress Checks are to be used for Formative assessments only.

Year 2, Day #20	Class #12 of Year 2 Unit 2

20.1 Methods Exam

- See Methods Final

Unit 3 - Arrays and Lists - 15 Classes

In this lesson plan doc, this unit will often be referred to as Data Structures. For AP CS Principles a dynamic data structure is used. For this unit, we start with a static data structure (Arrays) to focus on mastering the algorithms. The Java ArrayList is introduced later as it more closely mirrors the CS Principles structure.

Remember to insert the Unit Zero videos at various points throughout the lessons. In some cases, these can be used to supplement sort lessons. In other cases, it may extend the suggested length of the unit.

Unit Quick Links
Full Activities Folder
Direct Link to Teaching Examples folder (must be downloaded)
Student example assignments - Available to teachers who attend our PD sessions

Year 2, Day #21	Class #1 of Year 2 Unit 3

21.1 Introduce the concept of a Data Structure

- Search *Java Collections* and view and discuss the Collections subclass which contains most but not all
- There are 2 data structures that you will be introduced to in this course: Arrays and ArrayLists.
    - The use of lists allows multiple related items to be treated as a single value. Lists are referred to by different names, such as array, depending on the programming language.
- Both are often referred to as types of LISTS (Note: *List* in Java is a specific data structure that is not part of this course).
    - List procedures are implemented in accordance with the syntax rules of the programming language.
- Mostly focusing on using Arrays this unit with some intro to ArrayLists as they are closer the AP CSP List structure.
- Plus an introduction to saving to Files
- We will be looking briefly at searching and sorting
    - Linear search - very slow for large datasets
    - Binary Search - is WAY faster but only works with sorted data (sort techniques will be explored in a very cursory fashion)

21.2 Discuss Arrays as an abstraction of a cupboard

- Use the rows of the seats in the room as an analogy
- you must always ask: how many shelves, what will it hold (with the analogy being how wide do you need to make the cupboard)
- A string is stored as a list of characters in memory

21.3 Arrays Example 1

- Create a New Project called Data Structures to hold this Unit's work.
- Ex1_ArraysIntro.java
- Create two arrays, one that holds names and one that holds their wage
- How do we print out the third person's info
- Change that person's salary
- Create a hoursWorked array and calculate the pay for one person
- Use a single For Loop to print out all the info
- Mini-challenge: How could we add up the total salaries for the whole month
- Terminology to discuss:
    - A *list* is an ordered sequence of elements. An *element* is an individual value in a list that is assigned a unique index.

- The use of lists allows multiple related items to be treated as a single value. Lists are referred to by different names, such as *array*, depending on the programming language.
- An *index* is a common method for referencing the elements in a list or string using natural numbers.
- In Java, arrays and ArrayLists use an index starting at ZERO. In the CSPrinciples List, they start at ONE.

21.4 Arrays Example 2 - Bar Chart

- See Ex2_BarChart.java

Year 2, Day #22	Class #2 of Year 2 Unit 3

22.1 Impact of Data Video (Lesson 1 of 3) - Big Data Revolution Documentary

- This excellent hour long documentary is **Highly Recommended.**
- The video's content relates directly to the key concept of Models and Simulations plus gives a nice introduction to privacy and personal data issues.
- It is loosely split into chapters as follows:
    - Introduction (until 4:05)
    - Data Revealing Invisible Worlds (until 14:09)
    - Data watching over us (until 25:51)
    - Data Creating Intelligent Systems (until 36:55)
    - Data The Future of Revolution (until 38:35)
    - Data Targeting You (until 41:08)
    - Data The Dark Side (until 45:50)
    - Data Connecting the World (until end of video 52:22)

- **Today, watch up to the end of Data Revealing Invisible Worlds** (until 14:09)
- Use this portion of the video to discuss the nature of data
- Discuss the ideas presented in the video about how information and data has evolved including these AP CSP Essential Knowledge Statements:
    - *Information* is the collection of facts and patterns extracted from data.
    - Data provide opportunities for identifying trends, making connections, and addressing problems.
    - Digitally processed data may show correlation between variables. A correlation found in data does not necessarily indicate that a causal relationship exists. Additional research is needed to understand the exact nature of the relationship.
    - Programs are used in an iterative and interactive way when processing information to allow users to gain insight and knowledge about data.
    - Programmers can use programs to filter and clean digital data, thereby gaining insight and knowledge.
    - Combining data sources, clustering data, and classifying data are parts of the process of using programs to gain insight and knowledge from data.
    - Insight and knowledge can be obtained from translating and transforming digitally represented information.
    - Patterns can emerge when data are transformed using programs.

22.2 Tools for filtering, cleaning, and processing data

- Discuss that datasets provide challenges. Discuss the following AP CSP Essential Knowledge Statements:

    DAT-2.C.2: Data sets pose challenges regardless of size, such as:
    - the need to clean data
    - incomplete data
    - invalid data
    - the need to combine data sources

    DAT-2.C.3: Depending on how data were collected, they may not be uniform. For example, if users entered data into an open field, the way they choose to abbreviate, spell, or capitalize something may vary from user to user.

    DAT-2.C.4: Cleaning data is a process that makes the data uniform without changing its meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word).

DAT-2.C.7: Large data sets are difficult to process using a single computer and may require parallel systems.
DAT-2.C.8: Scalability of systems is an important consideration when working with data sets, as the computational capacity of a system affects how data sets can be processed and stored.

- Tools exist that allow people to clean and analyze data including Speadsheets like Excel and Google Sheets, but also more advanced tools.

- Tour the students through some tools for cleaning, analyzing, and presenting data
    - *OpenRefine* (formerly Google Refine) demonstrates how messy data needs to be cleaned. https://openrefine.org/
        - Watch the video called “Clean and Transform Data” embedded on the above page.
        - OpenRefine is implemented in Java and so is a good example of a real Java application.
    - *Jupyter*: A free tool for Data Science https://jupyter.org/
    - *Tableau* is another popular tool for analyzing data sets and include a free version
- The above tools can also be extended with a bit of programming ability using Java, Python or other languages.

22.3 Extracting Information from Data
Give students 10 minutes to explore a data set from Edmonton (or many cities) Open Data Portal and open it in Excel or Google Sheets
    - https://data.edmonton.ca/
    - https://www.toronto.ca/city-government/data-research-maps/open-data/
Then in partners, have them discuss the following:
- What dataset did you find interesting
- What information could you extract from this data (what statistics could you derive? what conclusions could you hope to make by analyzing this data more closely?)
- Would there be need to clean the data?

22.4 Arrays Example - Door prize generator
- Ex3_DoorPrize.java
- Introduces a printArray method that takes in an array as a parameter
- Store 5 names in a String Array
- print them all using a For Loop
- use Math.random to choose a door prize winner
- add a phone number array and print that info too.

22.5 Data Structures Assignment 1 - Superheroes
- See doc in Resources Folder
- Use it more as learning activity than an assignment.

Year 2, Day #23	Class #3 of Year 2 Unit 3

23.1 Impact of Data Video (Lesson 2 of 3) - Big Data Revolution Documentary

- Introduce part 2 of the video by reviewing the sections on visualization.
    - Programming languages such as Processing are good for creating visuals but often other tools dedicated tools can do the job more simply and/or be leveraged with a custom written program
    - Discuss how audio recordings of the family at 11:09 are transformed into graphs at 13:22
        - Lesson Plan Year 2, Page 36, Activity 23.1, the recordings over time are **clustered** based on time and location
        - Programs, just like we write in Processing and Java, create the graphs and visuals shown in the video.
    - Show data combining/extraction at time 1:17 of *unnamed soundsculpture - documentation-* https://vimeo.com/38874664
    - Programs can be used to process data to acquire information.
    - Tables, diagrams, text, and other visual tools can be used to communicate insight and knowledge gained from data.
    - Search tools are useful for efficiently finding information.
    - The size of a data set affects the amount of information that can be extracted from it.
    - Data filtering systems are important tools for finding information and recognizing patterns in data.
    - Programs such as spreadsheets help efficiently organize and find trends in information.

- Watch sections Data watching over us (until 25:51) & Data Creating Intelligent Systems (until 36:55)
- Discuss the use of Simulations to make predictions about the future including these AP CSP Essential Knowledge Statements
    - Models and simulations are used to generate new understanding and knowledge.
    - Simulations are abstractions of more complex objects or phenomena for a specific purpose.
    - A *simulation* is a representation that uses varying sets of values to reflect the changing state of a phenomenon.
    - Simulations often mimic real-world events with the purpose of drawing inferences, allowing investigation of a phenomenon without the constraints of the real world.
    - The process of developing an abstract simulation involves removing specific details or simplifying functionality.
    - Simulations can contain bias derived from the choices of real-world elements that were included or excluded.
    - Simulations are most useful when real-world events are impractical for experiments (e.g., too big, too small, too fast, too slow, too expensive, or too dangerous).
    - Simulations facilitate the formulation and refinement of hypotheses related to the objects or phenomena under consideration.
    - Random number generators can be used to simulate the variability that exists in the real world.

- After watching the video, have the students discuss the following in partners or small groups and then share their answers with the whole class:

-
- What is an example of a PROGRAM discussed in the video that was used to gain insight and knowledge from data? Explain.
- What are 2 examples from the video of how computers were used to simulate (and therefore abstract) real world phenomena or outcomes? Explain how.
    - Sample answers could include the Flight traffic visualization, the New Years tweets.
- Are games simulations? Why or why not?
- What are the similarities between these simulations and their real world contexts.

23.2 Ex3_BasicAlgorithms.java

- This example also uses a new static class: ArrayUtil.java to take care of the printing
- Discuss that Algorithms are used to extract and modify data. Many more common tasks have established algorithms. Extracting and modifying data includes:
    - Knowledge of existing algorithms can help in constructing new ones
    - transforming every element of a data set, such as doubling every element in a list, or extracting the parent's email from every student record
    - filtering a data set, such as keeping only the positive numbers from a list, or keeping only students who signed up for band from a record of all the student
    - combining or comparing data in some way, such as adding up a list of numbers, or finding the student who has the highest GPA
    - visualizing a data set through a chart, graph, or other visual representation
- This runs through a variety of array processing algorithms.
- Discuss/review the following concepts:
    - Traversing a list can be a complete traversal, where all elements in the list are accessed, or a partial traversal, where only a portion of elements are accessed.
    - Iteration statements can be used to traverse a list.
    - Knowledge of existing algorithms that use iteration can help in constructing new algorithms.

23.3 Assignment 2 - Storing Marks

- See doc in Resources folder

Year 2, Day #24	Class #4 of Year 2 Unit 3

24.1 Impact of Data Video (Lesson 3 of 3) - Big Data Revolution Documentary

- Introduce the remainder of the video by reviewing the sections previously seen.
- Talk about the inherent issues surrounding data collection to increase the likelihood that the simulation will have its intended effects
    - Often, a single source does not contain the data needed to draw a conclusion. It may be necessary to combine data from a variety of sources to formulate a conclusion.
    - The ability to process data depends on the capabilities of the users and their tools.
    - Data sets pose challenges regardless of size, such as:
        - the need to clean data
        - incomplete data
        - invalid data
        - the need to combine data sources
    - Depending on how data were collected, they may not be uniform. For example, if users entered data into an open field, the way they choose to abbreviate, spell, or capitalize something may vary from user to user.
    - Cleaning data is a process that makes the data uniform without changing its meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word).
    - Large data sets are difficult to process using a single computer and may require parallel systems.
    - Scalability of systems is an important consideration when working with data sets, as the computational capacity of a system affects how data sets can be processed and stored.
- Data can contain biases such as gender or race
    - Google Image Search Bias
        - https://www.huffingtonpost.ca/entry/google-image-gender-bias_n_7036414
    - Algorithms of Oppression
        - https://nyupress.org/9781479837243/algorithms-of-oppression/
- Many of the darker side of data issues will have been discussed in earlier classes. Take this time to reiterate some of the key concepts around personal information protection including these AP CSP Essential Knowledge Statements.
    - Personally identifiable information (PII) is information about an individual that identifies, links, relates, or describes them. Examples of PII include:
        - social security number
        - age
        - race
        - phone number(s)
        - medical information
        - financial information
        - biometric data
    - Search engines can record and maintain a history of searches made by users.
    - Websites can record and maintain a history of individuals who have viewed their pages.
    - Devices, websites, and networks can collect information about a user's location.
    - Technology enables the collection, use, and exploitation of information about, by, and for individuals, groups, and institutions.
    - Search engines can use search history to suggest websites or for targeted marketing.
    - Disparate personal data, such as geolocation, cookies, and browsing history, can be aggregated to create knowledge about an individual.
    - PII and other information placed online can be used to enhance a user's online experiences.

- PII stored online can be used to simplify making online purchases.
- Commercial and governmental curation of information may be exploited if privacy and other protections are ignored.
- Information placed online can be used in ways that were not intended and that may have a harmful impact. For example, an email message may be forwarded, tweets can be retweeted, and social media posts can be viewed by potential employers.
- PII can be used to stalk or steal the identity of a person or to aid in the planning of other criminal acts.
- Once information is placed online, it is difficult to delete.
- Programs can collect your location and record where you have been, how you got there, and how long you were at a given location.
- Information posted to social media services can be used by others. Combining information posted on social media and other sources can be used to deduce private information about you.

24.2 Continue Ex3_BasicAlgorithms.java

- Review with some similar algorithms that use different filtering criteria

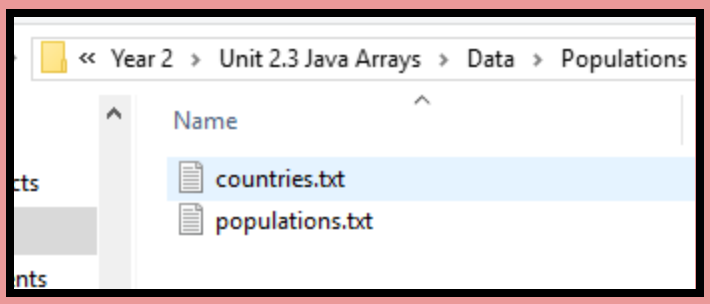
24.3 Introduce Assignment 3 - Guitar Store Revisited

- Part G (Printing out the bill) is surprisingly difficult. You may need to help walk them through it.

Year 2, Day #25	Class #5 of Year 2 Unit 3

25.1 Introduction to using Files for Long Term Storage

- Uses 2 java files plus some data files stored in the project folder:
  - Files.java
  - Ex4_Country_Files.java
  - Countries.txt & populations.txt
- See the image to the right for the file structure for saving the .txt files.
- I usually give them a partially completed Files.java to start and discuss as I add some details. I also provide the data files



- Once the data is loaded and tested, continue through Ex4_Country_Files.java which once again reviews some common algorithms

25.2 Introduce Assignment 4 Bike A Thon algorithms

- The Bike-A-Thon is our major school fundraising event and thus is a good topic for this assignment. You might need to customize the content to appeal to your students.

  Bike-a-thon Data Structures - Assignment 4

Year 2, Day #26	Class #6 of Year 2 Unit 3

26.1 Work time on all past Arrays assignments.

26.2 Video: Algorithms and what they can not solve: P vs NP - https://youtu.be/YX40hbAHx3s (11 minutes)

- Note: This video is very challenging and may only be approachable by more academic students.
- Before starting, discuss problems and as an example, the travelling salesman problem?
    - A problem is a general description of a task that can (or cannot) be solved algorithmically. An instance of a problem also includes specific input. For example, sorting is a problem; sorting the list (2,3,1,7) is an instance of the problem.
- This is another intellectual video but quite interesting and it leads well into a discussion of the following AP concepts on which you may want students to take notes.
    - Reasonable Time - the number of steps that an algorithm takes is less than a polynomial function of the input.
        - Where n is the number of items of data:
        - $4 * n^2$ is reasonable vs 4^n is not as this is a polynomial function of n
    - Heuristics - a technique that allows for an approximate solution where no reasonable solution exists
- Decision Problems vs Optimization Problems
    - A decision problem is a problem with a yes/no answer (e.g., is there a path from A to B?).
    - An optimization problem is a problem with the goal of finding the "best" solution among many (e.g., what is the shortest path from A to B?).
- Decidable vs Undecidable problems
    - Undecidable Problem
        - may have instances that have an algorithmic solution but there is no solution for all instances of the problem
        - An undecidable problem is one for which no algorithm can be constructed that is always capable of providing a correct yes-or-no answer
        - An undecidable problem may have some instances that have an algorithmic solution, but there is no algorithmic solution that could solve all instances of the problem.
    - Decidable Problem
        - an algorithm can be constructed to answer yes or no for all inputs
        - An undecidable problem may have some instances that have an algorithmic solution, but there is no algorithmic solution that could solve all instances of the problem.

26.3 Analyzing Algorithms Student Activity

- This assignment reinforces the topics covered in this and other videos. It can be completed in partners or individually.
- Found in folder 2.3 - Year 2: Java Arrays & Lists

Year 2, Day #27	Class #7 of Year 2 Unit 3

27.1 Linear Search Algorithm

- Add notes in a comment
    - AKA Sequential Searching
    - Linear searching is slow but it is the only search algorithm that we will discuss that works on unsorted data (remind them about the library algorithms example from class #1)
    - Checks each item in the list, in order, until the desired item is found or the list is exhausted.
    - Designed to find the only (or FIRST) match but can be adapted to find multiple results
    - Can be adapted to use .contains or .startsWith, etc.
    - Returns the index value to look up other data
    - Often returns a negative number to indicate not found

- Within the main method only, but discuss the method when done
- Use the Files class to work with the NHL Statistics to load names, teams and something else we have not used

- Create a foundIndex variable. Set it to -1 to begin with.
- Search for a name and print the stats for that player
- Use a <0 for not found.

- Search for a name and trade that player to the “Golden Coupries”
- Reuse the foundIndex variable but discuss if you might need more than one.
- This time, use a while loop instead

- Search for a name and update their games played.
- Search for Number of Games Played for >82 - state the season is over. Remind them that it only finds the first one.

27.2 Work time on all past Arrays assignments.

Year 2, Day #28	Class #8 of Year 2 Unit 3

28.0 Tracing Code Activity 4

- Walk students through the Exam syntax and give let them test their understanding with Khan Academy.

28.1 Array Methods Lesson

- In comment notes, explain the difference between pass by value and pass by reference

- Demonstrate by recreating the NHL files example and creating a method that changes the team name of a few of the cities (CHI, DET, EDM to their full team names).
    - As some players are listed under two teams COL/MTL, use String.contains and String.replace

- Create a Linear Search Method for last names
- Copy it to our ArrayUtilities class and test it on the Countries example

28.2 Assignment 5 Translator

- This is a relatively simple assignment to ensure mastery of Linear Search

Year 2, Day #29	Class #9 of Year 2 Unit 3

29.1 Introduce Assignment 7 Coffee Counter

- This is a major assignment in terms of marking value
- Discuss the use of the search method

29.2 Work time on assignments

Year 2, Day #30	Class #10 of Year 2 Unit 3

Note to AP CSP Teachers: Specific implementations of the binary search are outside the scope of the course and the AP CS Principles Exam but a quick discussion of the algorithm is required.

30.1 Introduce BinarySearch with an unplugged activity

- Line everyone up alphabetically by first name
- Number their array slot starting at zero
- Give Start and End sheets, calculate Mid start + end/2 and remind them about integer math
- First look for a name that exists. Say we will search until we find it.
- Then followed by one that does not BUT Get it wrong by not having an ending condition
- Discuss the following concepts:
    - The binary search algorithm starts at the middle of a sorted data set of numbers and eliminates half of the data; this process repeats until the desired value is found or all elements have been eliminated.
    - Data must be in sorted order to use the binary search algorithm.
    - Binary search is often more efficient than sequential/linear search when applied to sorted data.

30.2 Discuss algorithmic efficiency

- Why learn Binary Search when Lineary Search works?
- Efficiency is an estimation of the amount of computational resources used by an algorithm.
- Resources to consider include time, memory & processing power.
- An algorithm's efficiency is determined through formal or mathematical reasoning.
- An algorithm's efficiency can be informally measured by determining the number of times a statement or group of statements executes.
- Different correct algorithms for the same problem can have different efficiencies.

30.3 Binary Search Review Using AP Classroom

- If you have access to AP Classroom, take a few moments to go over the Topic Questions for 3.11 Binary Search. There were 10+ sample multiple choice questions available at the time this document was produced.

30.4 Work time on all assignments

- Challenge advanced students to find and use a binary search algorithm in place of a linear search where appropriate

Year 2, Day #31-33	Class #11-13 of Year 2 Unit 3

31.1 List Practice Questions Review Using AP Classroom

- If you have access to AP Classroom, take a few moments to go over the Topic Questions for 3.10 Lists. There were 20+ sample multiple choice questions available at the time this document was produced.

31.2 Introduce the Data Struct 8 Arrays Wrap Up Project

- This project generally takes 3-4 days of class time
- Students who have struggled to keep up should choose option 6 or 7 as they are far easier and will take less time.
- The first 2 have a really interesting A.I.esque extension for top students
- This assignment gives students open ended tasks, thus encouraging them to find creative solutions. Give them ideas on how to plan and design their program before beginning their coding process.
- This project focused on developing the following AP CSP Enduring Understandings, amongst others:
    - CRD-2 - Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/ feedback cycles, and that leaves ample room for experimentation and risk-taking.
    - AAP-2 - The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.

Year 2, Day 34	Class #15 of Year 2 Unit 3

34.1 Arrays Exam

- Note that the version in the Resources folder makes reference to 2D arrays which are not part of CSPrinciples and therefore not in these lessons. The test will need updating

Unit 4 - Introduction to Object Oriented Programming
5 Classes

Object Oriented Programming is not part of the AP CS Principles course but I feel it is a fundamental programming concept. If your local curriculum calls for a more robust treatment of OOP than this lesson doc offers, feel free to contact us for additional content.

Note that for OOP, I start to use Packages. These appear as subfolders in the Examples provided. I will use the format of *PackageName… JavaFileName.*

OR

CodeHS also has a strong Object Oriented Programming chapter in its AP Java unit. You might want to consider using it instead of my lessons for a change of pace.

OR

The Coding Train Youtube channel can be used to introduce OOP in Processing. “Coding Train” is produced on YouTube by the excellent Daniel Schiffman and is not related to us in any way. Daniel Schiffman’s book “Learning Processing” is also an excellent resource.

- Schiffman’s OOP Playlist

Unit Quick Links
Full Activities Folder
Direct Link to Teaching Examples folder (must be downloaded)
Student example assignments - Available to teachers who attend our PD sessions

Year 2, Day #35	Class #1 of Year 2 Unit 4

35.1 Intro to ArrayLists

- See the example code in ArrayLists...Ex1_ArrayLists.java
- Set up 3 parallel ArrayLists: String Coupriegram followers, String City of Residence, Integer ratings
- Walk through the basic ArrayList methods:
    - Add(object)
    - add(location,object)
    - set(location, object)
    - remove(location)
    - get(location)
- Compare these methods to the List functions in the AP CSP Course Description (starting on page 122.

- As a challenge: Print followers with ratings greater than ___

35.2 OOP Zero Assignment - ArrayLists

Year 2, Day #36	Class #2 of Year 2 Unit 4

36.1 Review ArrayList operations with a new example

36.2 Work time on ArrayLists Assignment

Year 2, Day #37	Class #3 of Year 2 Unit 4

37.0 Review Abstractions and data Abstraction Concepts Below

'Objects' in programming are a type of Data Abstraction

- AAP-1.D.1: Data abstraction provides a separation between the abstract properties of a data type and the concrete details of its representation.
- AAP-1.D.2: Data abstractions manage complexity in programs by giving a collection of data a name without referencing the specific details of the representation.
- AAP-1.D.3: Data abstractions can be created using lists.
- AAP-1.D.4: Developing a data abstraction to implement in a program can result in a program that is easier to develop and maintain.
- AAP-1.D.5: Data abstractions often contain different types of elements.

Then have students discuss the following with a partner.

- How can a pacman ghost and a character in Call of Duty both be considered abstractions of an enemy soldier?
- At the LOWEST LEVEL of abstraction, how is a soldier represented in a computer?
- Lists can be used to create Data Abstractions. Explain to each other how data abstraction manages complexity by each referencing one of our past List assignments.

37.1 Introduction to Object Oriented Programming

Discuss the idea of OOP as simply an organizational philosophy as a means of Data Abstraction. OOP has the following advantages:

No more Parallel Arrays!

- Remind them about the Coffee Counter assignment from last year. Discuss how OOP replaces the parallel arrays idea. Eventually our objects will not just be 'data classes' and will hold more sophisticated methods

Organization and Reusability of code - Such as the Files class
Sharing libraries - Such as Scanner,System, etc.
Scalability
Inheritance
Encapsulation and accessibility

37.2 Create a OOP Notes Google Doc

- Project
- Package
- Class
- Types of Classes (we will use this year)
    - Instance, Instantiation (including GUI classes)

        - This is a blueprint for creating instances.
        - Each instance
    - Static Class
    - Inherited Class
    - Main class

Static Class Discussion - Discuss

- Utilities classes such as the Files Class, Input class, Math. These are all static classes
- Useful, easily copied
- Accessed by making things PUBLIC and STATIC. You just put the name of the class followed by the method or variable names.

Instance Classes

- Act as blueprints for creating Instances of an object
- They are what Java was made for
- Use DOT notation to access methods and variables
- They are unaware of the existence of other Instances. (No search loops in here!)

Main Class

- The launcher for the program (including the launching of GUI when we get there)
- Often stores main Data Structures

37.4 OOP Example 1 - Refactor the followers example

- Ex1_Followers… Follower.java and CouprieGram_Main.java
- No ArrayList in this example
- Demo printing an object
- THEN add a toString method that prints what we want. This OVER RIDES the built in toString method that is called when you use .println(object)

37.5 Catch up time or add more methods to Followers

Year 2, Day #38	Class #4 of Year 2 Unit 4

38.1 Continue Notes Doc or just review verbally

Advantages of OOP

Organization and Reusability of code - Such as the Files class
Sharing libraries - Such as Scanner,System, etc.
Scalability
Inheritance
Encapsulation and accessibility (discuss private vs public briefly)

Using an Instance Class

Most instance classes have 3 components:

- Instance Variables declaration Area
- CONSTRUCTOR method - to initialize the variable
- Instance Methods - affect this instance only

How

- You create an IDENTIFIER in main using the NEW keyword
- This runs the constructor method
- Identifiers can be unique names or slots in a data structure

Reiterate the use of Libraries must be done legally.

- Some examples of legal ways to use materials created by someone else include:
  - Creative Commons—a public copyright license that enables the free distribution of an otherwise copyrighted work. This is used when the content creator wants to give others the right to share, use, and build upon the work they have created.
  - open source—programs that are made freely available and may be redistributed and modified
  - open access—online research output free of any and all restrictions on access and free of many restrictions on use, such as copyright or license restrictions
- The use of material created by someone other than you should always be cited.
- Creative commons, open source, and open access have enabled broad access to digital information.

38.2 OOP Example 2 with the Monster Class

- Ex2_MonsterGame… Monster.java and Monster_Main.java
- On the next page is some starter code for the Monster class. Have them copy the code on the next page before continuing.
- Finish off the Montster constructor and instance methods
- In main, create an instance of the Monster class and Print the results of all methods
- Notice the public methods that pop up once you create an instance. Discuss the use of private and public variables

Monster Start Code - Provided to Students

```
package Ex1_Monster;

public class MonsterV2 {
    public String type;
    public boolean isEvil;
    public int strength;
    public int health;
    public String description;

    //************* constructor methods *************************
    public MonsterV2(String t, boolean e, int s){



        String[]sizes = {"small","big","enormous"};
        String[]colors = {"green","blue","violet"};
        String[]look= {"ugly","polkadotted","furry"};
        description = sizes[(int)(Math.random()*3)] + ", " +  colors[(int)(Math.random()*3)]+", " + look[(int)(Math.random()*3)] ;

    }

    //**********  Instance methods  *********************
    public int attack(){
        //when the monster is attacking
        //returns the amount of damage done to the player


        return 0;  //temp line to avoid error
    }//end attack

    public int defend(int damage){
        //when the monster is defending
        //returns the current health of the monster


        return 0;  //temp line to avoid error
    }//end defend



}
```

End of Provided OOP Lessons

Note that there is more OOP example code and assignments in the provided materials. If you have time and interest, please contact me and I would be happy to provide more resources.

CodeHS also has a strong OOP unit for introducing the concepts in Java.

The Coding Train Youtube channel can be used to introduce OOP in Processing. Coding Train is produced by the excellent Daniel Schiffman and is not related to us in any way. Daniel Schiffman's book "Learning Processing" is also an excellent resource. Schiffman's OOP Playlist

Unit 5 - CS Theory - 8 Classes

Most of the AP CSPrinciples materials will be covered in Year 1, through this year's Unit 0 videos (do not forget to show them) or through the Exam Prep unit still to come. This unit is therefore quite short but does cover some key content that will otherwise slip through the cracks.

For Alberta Teachers, I have significantly more CS Theory content elsewhere that is needed to cover the Computer Science 2 curriculum. Do not hesitate to request it if desired.

Remember to insert the Unit Zero videos at various points throughout the lessons. In some cases, these can be used to supplement sort lessons. In other cases, it may extend the suggested length of the unit.

Unit Quick Links
Full Activities Folder
Direct Link to Teaching Examples folder (must be downloaded)
Student example assignments - Available to teachers who attend our PD sessions

Year 2, Day #39	Class #1 of Year 2 Unit 5

39.1 Binary Review and New

- Discuss that we will use IP Addresses and Colors as a recurring example for binary and hex
- Review IP addresses and domain names (review with code.org video) and discuss the following major concepts on the Principles Exam
    - IP addresses are Hierarchical and that hierarchy in the DNS helps it to scale.
    - Binary is the lowest level of abstraction
    - Use the wikipedia page on IPV6 to review binary/hex
        - IPV6 Addresses are in hexadecimal
    - Optional: Practice converting an IPV4 address from decimal to binary

- Binary Addition
    - This Adding in Binary Khan Academy video (2:47) has a nice explanation
    - Adding binary numbers produces a SUM bit and a CARRY bit. (Later we will see how logic gates do the math).
    - In computers memory is limited. Plus, because languages limit the size of numbers in a variable, overflow or other errors are possible.
    - Optional - Discuss other applications of binary including the Parity Bit and Checksums for data transmission
        - Watch youtube video on Parity Bit? (11:00) or just discuss
    - Practice with some basic binary additions statements (see the Digital Circuits assignment for examples). 4 bit numbers would be sufficient and discuss what happens if we do not have an additional bit allocated for the carry out.

39.2 Other Binary Math

- Discuss the CS Principles Sample Exam Question pasted below
    - What is your first, thoughtful answer
    - With a neighbour, do a proof

1. A video-streaming Web site uses 32-bit integers to count the number of times each video has been played. In anticipation of some videos being played more times than can be represented with 32 bits, the Web site is planning to change to 64-bit integers for the counter. Which of the following best describes the result of using 64-bit integers instead of 32-bit integers?

   (A) 2 times as many values can be represented.

   (B) 32 times as many values can be represented.

   (C) 2^{32} times as many values can be represented.

   (D) 32^2 times as many values can be represented.

39.3 Discuss Bitrate and Latency

- Bitrate is the bandwidth of a system, measured in bps (BITS per second)
- Latency is the TIME elapsed between the transmission and the receipt of the request.
- Problem -What bitrate would you need to upload a 1 Gigabyte video to Youtube in 1 minute? Show your math.

39.4 Binary Conversion Review/Practice

- Play the binary bonanza game for a few minutes as a review of last year's concepts

39.5 Review of logic gates with Logic Gates

- Remind them of the basic gates from Year 1: AND, OR & NOT
- Complete the Lighting Challenge (explained below) using Logic Lab. This is a series of puzzles, give them 10ish minutes to try to solve them all.

  Lighting challenge

  1. *A hallway light has two switches on either end. When either switch is changed, the light changes state.*
  2. *A bedroom light comes on with a motion sensor (use a button) BUT has a switch that if turned off the light is always off (ignoring the motion sensor)*
  3. *A kitchen light has 3 switches. If any of them are on, the light is on.*
  4. *Overloaded circuit. If all 3 of the above lights are one at one time, the circuit gets overloaded and all three turn off (you may also want to add a blinking LED to warn you of the overload).*

39.6 Hardware Theory Lesson 1 - How Logic Gates can be used to build an ALU

- Crash Course Video #5 - ALU (11:10) - Full series link
- Warn them about the pace of the video, they will get used to it. We will watch many of the videos over the next few days

Year 2, Day #40	Class #2 of Year 2 Unit 5

40.1 Hardware Theory Lesson 2 - How RAM and Registers are built

- Crash Course Computer Science #6 - RAM and Registers (12:16) - Full series link
- These videos go beyond the required content but they provide a good foundation for Computing Science hardware knowledge.
- Warn the students that they do not need to follow every last detail but should attempt to appreciate the big picture: that computers are actually very simple machines.

40.2 Introduce Hexadecimal Numbers

- This Khan Academy video does a nice summary: Hexadecimal Number System (7:33)
    - Key message: Hexadecimal is used because it takes fewer BYTES of memory to store larger numbers

40.3 Practice Converting to and from Hexadecimal Lesson

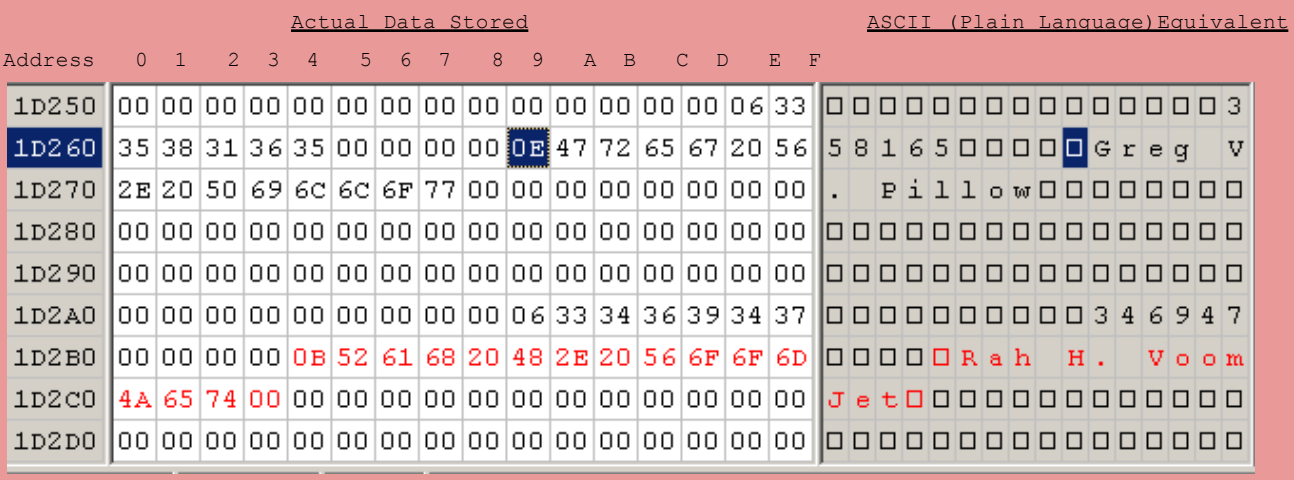
- Discuss direct conversion as shown in the video
- Then discuss the binary shortcut when going from Hex to Binary.
  A4B2
  A is 10 (1010)
  4 is 4 (0100)
  B is 11 (1011)
  2 is 2 (0010)
  Therefore the final Binary Number is: 1010 0100 1011 0010
- Practice converting numbers between the 3 number systems. Two Hexadecimal places will suffice.

40.4 Hexadecimal in Memory

- What does this actually look like in a computer? Show the Memory picture (below).
- Here is how to interpret the image:
    - In RAM, All data is actually stored as Hexadecimal numbers. The right side of the screen shows what the data looks like to humans. The left column and top row form a matrix of the actual memory addresses.
    - Remember that a STRING variable is stored in memory as a series of characters. The variable name will point to this memory location which is the start of the String. String.length() will return this number.
    - The number highlighted in location 1D269 (row 1D260 at bit 9), 0E means **14** in decimal. This byte indicates the *length* of the String “Greg V. Pillow” which

contains 14 characters.

- Go to ASCIITable.com and discuss. The letter ‘G’ is stored as number **47** according to ASCII Code and is located in 1D26A.



Remember that a STRING variable is stored in memory as a series of characters. The number highlighted in location `1D269` (row 1D260 at bit 9), `0E` means **14** in decimal. This byte indicates the *length* of the String “Greg V. Pillow” which contains 14 characters. The letter ‘G’ is stored as number **47** according to ASCII Code and is located in `1D26A`.

Year 2, Day #41	Class #3 of Year 2 Unit 5

41.1 Hardware Theory Lesson 3 - How the CPU is built

- Crash Course Computer Science #7 - The CPU (12:16) - Full series link

41.2 Introduction to Hexadecimal Numbers Assignment

- See doc in Resources folder
- This assignment usually takes about 30 minutes to complete

41.3 Digital Circuits Assignment Begins

- See doc in Resources folder
- This assignment usually takes about 2 hours to complete

Year 2, Day #42	Class #4 of Year 2 Unit 5

42.1 Optional Hardware Theory Lesson 4 - Continuing the Crash Course series

- The remaining videos tend to get complicated and beyond the scope of the AP curriculum but may be worth watching if your students are interested. - Full series link

42.2 Hex Conversion Practice

- If they feel they have mastered the concept, try the online FlippyBit game

42.3 Finish Digital Circuits Assignment

42.4 Optional: Hexadecimal Colours Assignment

- See doc in resources folder
- This activity is often a bit confusing but once they realize what to do (converting 12 Hex numbers to decimal) it usually only takes 30 minutes
- You may need to demo by completing the first square with them

Year 2, Day #43	Class #5 of Year 2 Unit 5

43.1 Ethical Issues in Computer Science
Note: Computer and Tech ethics should be a constant discussion throughout your course. This activity is designed to reinforce these discussions.

Go to the *Ethical Issues* folder here

- Make copies of all the documents (Shared Activities Folder… Year 2 Theory… Ethical Issues). This activity can be handled in a number of ways:
    - Used Shared Documents among all the students and control their access
    - Have students submit individual documents by choosing from the shared docs
    - Use the docs as discussion points rather than formal written work

Skim through each issue as a quick group discussion. There are 15 issues so this will take some time, even at the superficial level, so I sometimes split this up over the course of a few classes.

43.2 Discuss/review the following general concepts about Computer Ethics:

- Computing innovations can be used in ways that their creators had not originally intended:
    - The World Wide Web was originally intended only for rapid and easy exchange of information within the scientific community.
    - Targeted advertising is used to help businesses, but it can be misused at both individual and aggregate levels.
    - Machine learning and data mining have enabled innovation in medicine, business, and science, but information discovered in this way has also been used to discriminate against groups of individuals.
- People create computing innovations.
- The way people complete tasks often changes to incorporate new computing innovations.
- Not every effect of a computing innovation is anticipated in advance.
- A single effect can be viewed as both beneficial and harmful by different people, or even by the same person. Some of the ways computing innovations can be used may have a harmful impact on society, economy, or culture.
- Responsible programmers try to consider the unintended ways their computing innovations can be used and the potential beneficial and harmful effects of these new uses.
- It is not possible for a programmer to consider all the ways a computing innovation can be used.
- Computing innovations have often had unintended beneficial effects by leading to advances in other fields.
- Rapid sharing of a program or running a program with a large number of users can result in significant impacts beyond the intended purpose or control of the programmer.

43.3 Complete Issue Zero - Digital Divide together as an example:
Introduce the concept of the Digital Divide

- Bridging the Digital Divide (3:41) - focusing on the Digital divide within developed countries
- Digital Divide (4:10) - focusing on the Digital divide between rich and poor countries

Discuss/review the following AP CSP Essential Knowledge Statements:

- Internet access varies between socioeconomic, geographic, and demographic characteristics, as well as between countries.

- The “digital divide” refers to differing access to computing devices and the Internet, based on socioeconomic, geographic, or demographic characteristics.
- The digital divide can affect both groups and individuals.
- The digital divide raises issues of equity, access, and influence, both globally and locally.
- The digital divide is affected by the actions of individuals, organizations, and governments.

43.4 Model their upcoming assignment - For each issue, they are to do three things:

- Read the background info and the ethical question. The question is always related but to the background info but sometimes in an indirect way.
- Find a news article or commentary dedicated to the issue.
- Then provide the following information about the issue in general or about the specific ethical question:
    - Student names
    - New fact/statistic relate to the issue
    - Someone else’s opinion - preferably a direct quote
    - Your opinion(s)
    - Website source

43.5 Student Activity - Ethical Issues Rotation

Here is how I implement these lessons in my classroom:

- Students are paired up
- I assign each partnership 1 issue/document, assigning it by number. (Group 1 will complete Issue 1.) I used the ‘Shared Docs’ approach so there is a cell in a table that each group will use. IE, Group 1 will always put their answers into slot 1. Group 12 will always put their answers into slot 12.
- Students get exactly 15 minutes to complete their assigned issue.
    - They should be encouraged to negotiate and build consensus in their answers but could include separate answers if consensus is not reached.
- Groups then rotate to the next issue in order and again are given 15 minutes.
- (continued next class)

Year 2, Day #44	Class #6 of Year 2 Unit 5

44.1 Ethical Issues Rotation Continued - 30 to 40 minutes

- After their two assigned issues, groups are given time to complete 2 or 3 issues of their choice

Note to Teachers on the Ethical Issues Rotation activity
This activity is designed to provide a short introduction to the wide variety of issues that arise from today's tech. An alternative or reinforcing activity is to have students do a more indepth research of the issues that are brought forth by one computing innovation.

The **Analyzing Impact of Computing** activity that is part of the **AP Explore Teachers Resource** provides a good template for such a research project. If you did not complete it in Year 1, you should do so now.

44.2 Introduce the Software/Systems Development Lifecycle

- The SDLC is a model to plan the development of software or a system from start to finish
- *For this first lesson, just discuss the phases in generalities, specific notes will be a part of the next lesson.*
- A development process can be ordered and intentional, or exploratory in nature.
- There are multiple development processes. The following phases are commonly used when developing a program:
    - investigating and reflecting
    - designing
    - prototyping
    - Testing
- A development process that is iterative requires refinement and revision based on feedback, testing, or reflection throughout the process. This may require revisiting earlier phases of the process.
- A development process that is incremental is one that breaks the problem into smaller pieces and makes sure each piece works before adding it to the whole.

<u>5 Steps in our version of the SDLC</u>

Analysis - Determining the system's requirements/specifications including these AP CSP Essential Knowledge Statements:

- Program requirements describe how a program functions and may include a description of user interactions that a program must provide.
- Investigation in a development process is useful for understanding and identifying the program constraints, as well as the concerns and interests of the people who will use the program.
- A program's specification defines the requirements for the program.
- The design of a program incorporates investigation to determine its requirements.

- The purpose of computing innovations is to solve problems or to pursue interests through creative expression.
- An understanding of the purpose of a computing innovation provides developers with an improved ability to develop that computing innovation.
- Consultation and communication with users are important aspects of the development of computing innovations.
- Information gathered from potential users can be used to understand the purpose of a program from diverse perspectives and to develop a program that fully incorporates these perspectives.

Design - Planning the software and hardware

- In a development process, the design phase outlines how to accomplish a given program specification.
- The design phase of a program may include:
    - brainstorming
    - planning and storyboarding
    - organizing the program into modules and functional components
    - creation of diagrams that represent the layouts of the user interface
    - development of a testing strategy for the program

Implementation - Developing the software and hardware (usually done in a collaborative environment)

- Common models such as pair programming exist to facilitate collaboration.
- Effective collaborative teams practice interpersonal skills, including but not limited to: communication, consensus building, conflict resolution. Negotiation
- Effective collaboration produces a computing innovation that reflects the diversity of talents and perspectives of those who designed it.
- Collaboration that includes diverse perspectives helps avoid bias in the development of computing innovations.
- Online tools support collaboration by allowing programmers to share and provide feedback on ideas and documents.

Testing - Introduce for now. More to come below.

Deployment and Maintenance - Introduce for now. More to come below.

Year 2, Day #45	Class #7 of Year 2 Unit 5

45.1 Ethics Review Using AP Classroom

- If you have access to AP Classroom, take a few moments to go over the Topic Questions for 5.1 through 5.6. There were many sample multiple choice questions available at the time this document was produced. You may want to build a series of short quizzes out of these questions.

45.2 Software Development Life Cycle Notes

- See notes on the next pages. Create a template for students to fill in with some of the key words blanked out on each line.
- As an ongoing example, imagine your school district intends to create a new Grades Management system.

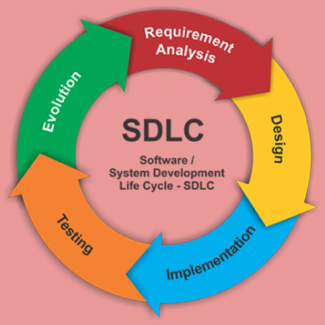
45.3 SDLC Assignment 1 - See Doc in Resources Folder

Full Notes: Software Development Life Cycle (aka Systems Development…)

- A way to organize the development of large scale projects

<u>5 Steps in our version of the SDLC</u>

1. Analysis
2. Design
3. Implementation
4. Testing
5. Deployment and Maintenance



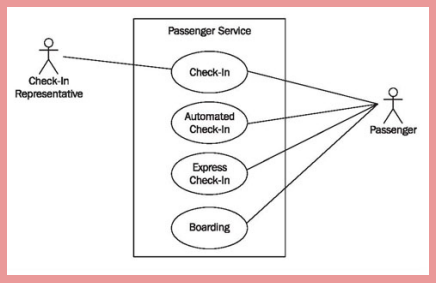
Analysis

Analysis involves and investigation to determine the program's requirements. This can be achieved through a combination of:

- Surveys, user testing, interviews, direct observation of end users in their normal environment, secondary research, etc

As part of this Analysis, the following will need to be considered:

- Will require hardware, software, both?
- What other systems (internal and external) will it need to interact with?
- What user groups are there and what are their needs?
- What are the problems with the old system?
- Identify which features are needed for each user group?

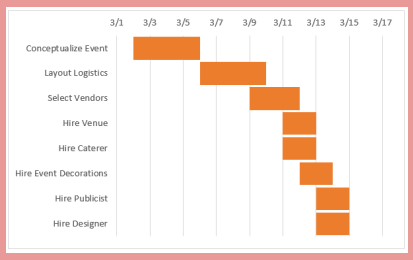


- *Brainstorm with neighbour, all the features that could be implemented that are needed by teachers. Then ask about new additions and subtractions for admin.Then families and students.*
- Use Case Diagram - stick figure diagram that explains all the features and which groups can use them (see image)
- Write up the contract

Design

- Plan from the coders/developers perspective

- What language? What data structures?
- The design phase of a program may include:
    - brainstorming
    - planning and storyboarding
    - organizing the program into modules and functional components
    - creation of diagrams that represent the layouts of the user interface
    - development of a testing strategy for the program
- Break the task down into teams, assign tasks, set timelines
- Gantt Chart - time management/planning tool (see image)



Implementation

- Writing of the code, working with hardware
- Debugging for many types of errors, including:
    - A *logic error* is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly.
    - A *syntax error* is a mistake in the program where the rules of the programming language are not followed.
    - A *run-time error* is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors.
    - An *overflow error* is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values.
    -
- Designing prototypes, interact with users, alpha testing

Testing

- Testing the near finished product or a new version during development.
- In the development process, testing uses defined inputs to ensure that an algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs.
- Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data.
- Program requirements are needed to identify appropriate defined inputs for testing.
- The following are effective ways to find and correct errors:
    - test cases

- hand tracing
- visualizations
- debuggers
- adding extra output statement(s)

Deployment and Maintenance

- How are you going to release your product?
- How are going to market your product
- Training?

Case Study -New Gradebook Software

Step	Activities and Documents
Analysis	
Design	
Implementation	
Testing	
Deployment & Maintenance	

Year 2, Day #46	Class #8 of Year 2 Unit 5

46.1 Systems Design - Sequential, Parallel vs Distributed Systems

Discuss the efficiency of various solutions using stock market trading and/or crypto-currencies as an example.

- *Sequential computing* is a computational model in which operations are performed in order one at a time.
- *Parallel computing* is a computational model where the program is broken into multiple smaller sequential computing operations, some of which are performed simultaneously.
- *Distributed computing* is a computational model in which multiple devices are used to run a program.

Discuss the benefits and challenges of the types of systems

- Comparing efficiency of solutions can be done by comparing the time it takes them to perform the same task.
- A sequential solution takes as long as the sum of all of its steps.
- Parallel computing consists of a parallel portion and a sequential portion. A parallel computing solution takes as long as its sequential tasks plus the longest of its parallel tasks.
- The "speedup" of a parallel solution is measured in the time it took to complete the task sequentially divided by the time it took to complete the task when done in parallel.
- Solutions that use parallel computing can scale more effectively than solutions that use sequential computing.
- Distributed computing allows problems to be solved that could not be solved on a single computer because of either the processing time or storage needs involved.
- Distributed computing allows much larger problems to be solved quicker than they could be solved using a single computer.
- When increasing the use of parallel computing in a solution, the efficiency of the solution is still limited by the sequential portion. This means that at some point, adding parallel portions will no longer meaningfully increase efficiency.

46.2 Student Activity: Sequential, Parallel and Distributed Systems Experiment

- Have students complete this 'experiment' in small groups. Note that it requires 20 stacking objects per group so you may want to run the groups 'sequentially'.
- Found in folder 2.5 - Year 2: Theory.

46.3 SDLC Assignment 2 - Locker 2040

- See Doc in Resources Folder
- This assignment has students model the SDLC for a futuristic device such as locker or bioinformatic device
- Students are allowed complete freedom, including suspending the laws of physics
- As written, it requires them to create a Use Case Diagram and a Gantt Chart (see notes on previous page). I recommend modelling a Gantt Chart using a spreadsheet first.
- It also discusses CrowdSourcing as a concept by giving students the option to create a Kickstarter style pitch video
- This assignment generally takes about 2 hours to complete.

Year 2, Day #47	Class #9 of Year 2 Unit 5

47.0 Computing in Parallel

- See this video about a parallel supercomputer at a national laboratory
    - What is a Supercomputer? https://www.youtube.com/watch?v=utsi6h7IFPs
- Emphasize that computers like this one are necessary for certain tasks like weather simulation.

47.1 Sequential, parallel and Distributed Computing - Quick Research Assignment

- For this assignment students are to focus on the banking industry. It is easily adaptable to other industries or to give students choice?
- Find an example of how the banking industry uses each of Sequential, Parallel and Distributed computing. Then for each:
    - What problem does each system solve
    - How are the problems similar? How are they different?
    - What is the relative efficiency of each
- Describe the benefits and challenges of parallel and distributed computing
- Which system is 'block chain'? How you anticipate it influencing the banking industry

47.2 Networks and Internet Review - worksheet part 1

In year 1, students were introduced to the basic workings of the Internet. This serves as a review and uses videos that up the level of technical knowledge presented.

The assignment is in two parts and contains links to the videos within but I recommend watching each video first and reiterating the concepts together, then assigning the worksheet.

- See the doc in the resources folder for the student worksheet

Video 1 - YouTube: Computer Networks: Crash Course Computer Science #28 (12:19)

- Related concepts including these AP CSP Essential Knowledge Statements:
    - A *path* between two computing devices on a computer network (a sender and a receiver) is a sequence of directly connected computing devices that begins at the sender and ends at the receiver.
    - Paths are often measured in Hops
    - *Routing* is the process of finding a path from sender to receiver.
    - The *bandwidth* of a computer network is the maximum amount of data that can be sent in a fixed amount of time.
    - Bandwidth is usually measured in bits per second.
    - Information is passed through the Internet as a *data stream*. Data streams contain chunks of data, which are encapsulated in *packets*.
    - Packets contain a chunk of data and metadata used for routing the packet between the origin and the destination on the Internet, as well as for data reassembly.
    - Packets may arrive at the destination in order, out of order, or not at all.
    - IP, TCP, and UDP are common protocols used on the Internet.
    - The Internet has been engineered to be fault-tolerant, with abstractions for routing and transmitting data.

- *Redundancy* is the inclusion of extra components that can be used to mitigate failure of a system if other components fail.
- One way to accomplish network redundancy is by having more than one path between any two connected devices.
- If a particular device or connection on the Internet fails, subsequent data will be sent via a different route, if possible.
- When a system can support failures and still continue to function, it is called *fault-tolerant*. This is important because elements of complex systems fail at unexpected times, often in groups, and fault tolerance allows users to continue to use the network.
- Redundancy within a system often requires additional resources but can provide the benefit of fault tolerance.
- The redundancy of routing options between two points increases the reliability of the Internet and helps it scale to more devices and more people.

How Networks and the Internet Work - worksheet part 1

- Have the students complete the first half of the worksheet based on video 1

47.3 In-School Field Trip - Map the Network

- Arrange to visit your school's server room and/or get support from your school's network administrator
- Trace the number of hops it takes to get from a student device out to the server (switches along the way)
- Discuss what additional hops it takes at the district level before reaching the school's ISP and the Internet backbone
- Optional: have students create a visual demonstrating these hops

Year 2, Day #48	Class #10 of Year 2 Unit 5

48.1 Review Using AP Classroom

If you have access to AP Classroom, take a few moments to go over the Topic Questions for 4.3 including Parallel and Distributed Computing. There were 30+ sample multiple choice questions available at the time this document was produced.

48.2 Networks and Internet Review Assignment Part 2

48.3 Video 2 - YouTube: The Internet: Crash Course Computer Science #29 (11:57)

- Related concepts:
  - The Internet is a computer network consisting of interconnected networks that use standardized, open (nonproprietary) communication protocols.
  - Access to the Internet depends on the ability to connect a computing device to an Internet-connected device.
  - A *protocol* is an agreed-upon set of rules that specify the behavior of a system.
  - The protocols used in the Internet are *open*, which allows users to easily connect additional computing devices to the Internet.
  - Routing on the Internet is usually dynamic; it is not specified in advance.
  - The *scalability* of a system is the capacity for the system to change in size and scale to meet new demands.
  - The Internet was designed to be scalable.

48.4 Traceroute in Class Discussion

- Use the traceroute program (explanation in the video description on youtube) and compare the ping times for multiple websites
- Find two to focus on where the physical distance is greater but the ping time is shorter
- In partners, have them discuss reasons for or causes of the disparity

48.5 How Networks and the Internet Work - worksheet part 2

- Have the students complete the remainder of the worksheet based on video 2

Year 2, Day #49	Class #11 of Year 2 Unit 5

49.1 Protecting Personal Information when Online - Using the *Explore Teachers Resource*

- These concepts have cropped up regularly throughout the courses but are always worth reviewing. It will also serve as a review of the ethics concepts earlier in the unit.
- Complete the *Collecting Data* task from the *Explore Teachers Resource*. Note that the aboutmyinfo.org site used only works fully within the United States. (I hope to find a Canadian equivalent soon.)

49.2 Review activities for Theory Exam

Year 2, Day #50	Class #12 of Year 2 Unit 5

50.1 Theory Unit Exam

- See doc in resources folder
- Review and practice of the logic gates components may be necessary before the test.

Unit 6 - The AP Create Task - 12 to 15 Classes

The AP Create Task is to be completed by students and submitted to the AP Digital Portfolio. It is a program written by students. Students can complete this in groups but the written responses and video must be completed individually.

AP requires that you give a minimum of 12 class hours (720 minutes) to this project. I generally give 12-15 classes resulting in as much as 20 hours of class time.

Unit Quick Links
Full Activities Folder
Direct Link to Teaching Examples folder (must be downloaded)
Student example assignments - Available to teachers who attend our PD sessions

Major Update: AP Exam and Create Task Changes for the 2023/24 School Year

In late August, the College Board announced a significant change to the Create Task and to the CSP Exam. The College Board will be providing additional PD for teachers in the fall and we will update our resources after we have participated in that training.

Updated Aug. 28, 2023.

Create Task Info for Teachers

The Create Task is designed to encourage students to use their creativity. As such, they should be free to choose from any programming tool they have experience in but:

- It is a SUMMATIVE task. They should not be using this time to learn a new tool/language.
- They will be required to identify Algorithms and Abstractions in their code. Their choice of tool must therefore allow them to write functions, loops, conditional statements, etc.

For AP, you are required to provide a minimum of 12 hours for students to produce 3 items:

- A video of your program running
- Written Responses

- Full Program code

I also require my students to complete a rigorous planning process and self evaluation plus present their program to the class.

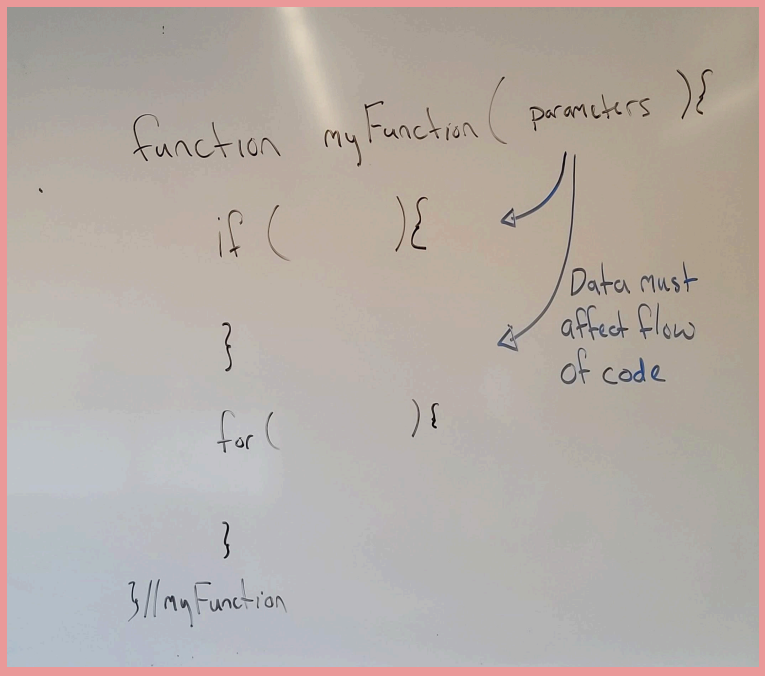
There are a variety of ways you can implement the Create Task but you must keep in mind the following key points:

- Students may fully collaborate on the program but I recommend students assign group members specific features/modules so that they limit their dependency on others.
- Students may NOT collaborate on the video and written responses
- Students must carefully document/source any code not written by them
- The teacher is VERY limited in how much they can aid the students in this task. See the Course Description for specifics

Core Programming Components of the Create Task

There are only 3 things that must be part of the Create Task code written by students:

- It must include input and output
- The use of a list
- A student written function that fulfills the following criteria:
    - It takes in at least one parameter
    - It contains a loop
    - It contains a selection statement (likely an IF statement) that is affected by the data passed in via the parameter. That is, students must be able to articulate how the flow of the code within the function can differ depending on the state of the parameter(s).



College Board Videos to Support you in the Create Task

The College Board has recently produced a number of really useful videos and webinars about the Create Task. The following descriptions and links are taken directly from the CSP Exam page of AP Central.

- **Watch Recorded Webinar: Evaluating the Create PT Like an AP Reader -** To gain additional skills and knowledge on how to read and score AP CSP student work like an AP reader, we invite you to watch the recorded webinar.
- **Student Webinar: Tips for Completing the Create Performance Task -** Get students to watch the recorded webinar for tips on planning and completing the Create performance task. We will also review common mistakes students make.
- **Teacher Webinars -** Watch webinar recordings on Create performance task updates, digital portfolio updates, and course and exam updates.
- **Create Performance Task Walk-Through -** Watch the recording of the Create Performance Task Walk-Through webinar, for guidance on task directions, task guidelines, and the submission process.

Subtask 1 - Introduce the Create Task

- Provide the handout from pages ??? of the CSP Course and Exam Description
- Read through the handout in detail
- Explain your own class specific procedures and marking scheme. The Create Task Summary Doc for Students in the Resources folder has an example.
- Give students lots of heads up.
- Have them tour through the sample projects in the Examples Folder. Remind them, however, that my students may have had substantially more time and/or access to different tools than yours will. Use these examples to get their creative juices flowing rather than projects to strive for.

Subtask 2 - Analysis and Proposal (Planning Doc #1)

- This doc is in the Resources folder
- There is also a completed student exemplar in the Resources Folder
- It requires them to formally propose their project.
- It also asks them to identify 'user groups' and create a Use Case diagram for their project. If you have not used Use Case Diagrams earlier in the course, you should skip this component.
- Recommend 1 hour (2 hours if creating a Use Case Diagram)

Subtask 3 - Design and Evaluation Planning (Planning Doc #2)

- This doc is in the Resources folder
- There is also a completed student exemplar in the Resources Folder
- Recommend 1 hours

There are 3 components to this doc

Marking Criteria

- This doc requires them to design their own marking criteria.
- I take a lot of time explaining what is an appropriate level of complexity and how to craft criteria. Generally criteria should be mostly objective.
- As this is a self-directed project, the majority of their final grade will be based on this criteria
- Students will be given opportunities throughout the project to revisit this criteria. See the Year 1 End of Year Project for details (as they are similar and not repeated here).

Gantt Chart

- I highly recommend using a Gantt Chart to plan their time, even if you have not used such a tool in class before.
- See the Student Exemplar
- I set it up with them and then give time to fill in the details.

Design Phase Documents

- The UML Diagram is certainly an optional component but I do recommend you require some sort of User Interface Sketch and/or Flow charts

Subtask 4 - The Coding

- Recommend 8-12 class periods
- Have the students reflect on their Gantt Charts and revisit their marking criteria at various points
- Once about 75% of the class time is completed, their criteria is locked-in

Subtask 5 & 6 - Creating the Video and Written Responses

- See the doc called Create Task Video and Written Responses in the Resources folder for a complete walkthrough of the requirements and suggested order.
- Tour them through the officially published sample submissions and how they were scored.
- Recommend 4 hours for completing all the written submissions plus video production time
- Note that the creation of the Full Program Code component often takes longer to gather and properly annotate than expected.

Subtask 7 - Final Edit and Prepare Official Submission Template

- They must be sure to use the official submission template. These are updated each year and therefore should be downloaded directly from the AP website.
- Tour them through the Create Task Rubric once again and have them edit, update and prepare update their responses.
- Pay close attention to the word limits.
- Recommend 1 hours

Subtask 8 - Self Evaluate Create Task

- You are not permitted to give the students any feedback until after the May deadline so I have them self-evaluate their submission based on their own criteria.
- See the Create Task Project Self Evaluation doc in the Resources Folder
- Recommend 1 hour

Subtask 9 - Submit Create Task

- I recommend they submit their Create Task to the AP Digital Portfolio as soon as finished, then review their submission at some point in the future before the deadline. This will allow you to browse their submissions and ensure compliance.
- At some point before the May deadline, they must indicate in the AP Digital Portfolio that their submissions are ‘FINAL’.

- The Create Task Video and Written Responses doc reiterates the details before they submit.
- Recommend 0.5 hours

Unit 7 Exam Prep - 9 Classes

This unit is designed to prepare students for taking the AP CSP End of Course exam. It involves review activities, shoring up small items that may have been missed and practice test writing.

I also sometimes incorporate final submission review (of the Explore and Create Tasks) into this unit. I have not included such time in this lesson doc.

Two outside resources are used: CodeHS (this will require a teacher account - the free version will be completely sufficient) and Khan Academy.

Note that I am not allowed to share official AP documents including the officially released practice exam. Once you have completed the Course Audit process, you will have access via the AP Website.

Unit Quick Links
Full Activities Folder
Direct Link to Teaching Examples folder (must be downloaded)
Student example assignments - Available to teachers who attend our PD sessions

Major Update: AP Exam and Create Task Changes for the 2023/24 School Year

In late August, the College Board announced a significant change to the Create Task and to the CSP Exam. The College Board will be providing additional PD for teachers in the fall and we will update our resources after we have participated in that training.

Updated Aug. 28, 2023.

Year 2	Class #1 of Year 2 Unit 7

7.1.1 Curriculum Analysis Doc (part 1)

- See separate doc in the folder: 2.7 Year 2 - Exam Prep
- Firstly, this document calls the students attention to the information about the exam itself.
- Secondly, it asks them to search the details of the Course Description document looking for things on which they need further review.
- It will likely take about 2 hours if done well. This is a dry but important activity so you may want to split it over a couple of classes. (part 1 today, part 2 next class)

For your information (and it is one of the questions in the above activity):

Big Ideas	Exam Weighting
Big Idea 1: Creative Development	**10–13%**
Big Idea 2: Data	**17–22%**
Big Idea 3: Algorithms and Programming	**30–35%**
Big Idea 4: Computer Systems and Networks	**11–15%**
Big Idea 5: Impact of Computing	**21–26%**

7.1.2 Explore the Exam Reference Sheet

- Go through the exam reference sheet (Appendix 1 of the CED) and look for differences/surprises
- Pay particular attention to:
    - Differences in assignment and other math operators such as mod and random
    - Use of Repeat and Repeat Until
    - List assignment and other list functions
    - Lists start at ONE not at zero
    - For Each Loop
    - Display & Input

7.1.3 AP Classroom Practice Quiz - Big Idea 3: Algorithms and Programming

- At this time, your students should be ready to write some practice exam questions for this Big Idea 3 but likely only for the Programming side. They will do well on the Programming questions once they get used to the syntax of the exam but may do not so well on the Algorithms side. A review activity for Algorithms is coming up soon.
- Create and administer a Formative Performance Task (or as I would call it, a practice quiz) on AP Classroom
- How to videos are available on the AP Classroom site

If you do not have access to AP Classroom, use CodeHS for resources or skip down to lessons below that make reference to Khan Academy reviews.

Year 2	Class #2 of Year 2 Unit 7

7.2.1 CSP Word of the Day: Scalability

- Each day, I take a few minutes to introduce a term that might crop up on the exam.
- Sometimes we discuss the definition, sometimes I have them look it up.
- In most cases, we search the Course Description and find the specific EK statements where it appears.
- This often leads the discussion to other related terms too.
- Today's Word: Scalability -The scalability of a system is the capacity for the system to change in size and scale to meet new demands. The Internet was designed to be scalable.
- Related terms: protocols, IP, TCP/IP, routers and packets

7.2.2 Finish Curriculum Analysis

7.2.3 AP Classroom Practice Quiz - Big Idea 1: Creative Development

- At this time, your students should be ready to write some practice exam questions for this Big Idea.
- Create and administer a Formative Performance Task (or as I would call it, a practice quiz) on AP Classroom
- How to videos are available on the AP Classroom site
- If you do not have access to AP Classroom, use CodeHS for resources or skip down to lessons below that make reference to Khan Academy reviews.

7.2.4 CSP Review Assignment 2 - Internet

- Note that the first two Review assignments are deliberately out of order.
- Instructions for this assignment can be found in the folder: 2.7 - Year2: Exam Prep

This activity makes use of the great Khan Academy CSP Review course.

Year 2	Class #3 of Year 2 Unit 7

7.3.1 CSP Word of the Day: Cleaning Data

- Cleaning data is a process that makes the data uniform without changing its meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word).
- Related terms: Data set, bias, filtering

7.3.2 Algorithms and what they can not solve

- If you have not watched this video earlier in the course, it does a good job of setting the stage for today's topics.
- Warning: This video is very challenging and may only be approachable by more academic students. P vs NP - https://youtu.be/YX40hbAHx3s (11 minutes)

7.3.3 CSP Review Assignment 1 - Algorithms

- Instructions for this assignment can be found in the folder: 2.7 - Year2: Exam Prep
- This activity makes use of the great Khan Academy CSP Review course.

Year 2	Class #4 of Year 2 Unit 7

7.4.1 CSP Word of the Day: Heuristics

- Review's yesterday's concepts
- Related terms: Reasonable time, Decidable and Undecidable problems
- Partnership discussion: What is a medical/bio-informatic problem for which we currently use a heuristic solution?
- Partnership discussion: Why is solving a Rubics Cube a reasonable time problem, but solving soduko not (when the puzzles scale up)?
- Partnership Discussion: Explain the existence of undecidable problems in computer science.

7.4.2 CSP Review Assignment 3 - Data Security

- Instructions for this assignment can be found in the folder: 2.7 - Year2: Exam Prep
- This activity makes use of the great Khan Academy CSP Review course.

7.4.3 Binary Numbers Review

- Have them play Binary Bonanza
- Take them to Topic 2.1 from the CED and review the Essential Knowledge statements related to binary and abstraction and errors such as overflow and round-off..
- Then discuss the answer to the following sample exam question.

1. A video-streaming Web site uses 32-bit integers to count the number of times each video has been played. In anticipation of some videos being played more times than can be represented with 32 bits, the Web site is planning to change to 64-bit integers for the counter. Which of the following best describes the result of using 64-bit integers instead of 32-bit integers?

   (A) 2 times as many values can be represented.

   (B) 32 times as many values can be represented.

   (C) 2^{32} times as many values can be represented.

   (D) 32^2 times as many values can be represented.

Year 2	Class #5 of Year 2 Unit 7

7.5.1 CSP Word of the Day: Intellectual Property

- Related terms: Creative Commons vs Open Source

7.5.2 Big Idea 4 - Computing Systems and Networks - Part 1 Create the Assignment

- See separate doc in Resources folder
- I call this the ‘Create an Assignment’ Assignment
- Here the students rewatch 1 video from the Code.org Internet video series used in Year 1 and create a test/assignment for others to complete in a future class.
- They do so in partners in my version.
- This activity will likely take the whole class

- You can either have the students print their assignments or compile them into a shared doc.
- Next class, have them complete the other students’ assignments.

Year 2	Class #6-8 of Year 2 Unit 7

Over the last few days of the course, have students write the Official Released AP practice exam on AP Classroom. This can be done at one time or over several classes. During that time, review the following topics if necessary.

7.6.1 Words of the Day

CSP Word of the Day: Intellectual Property
- Related terms: Creative Commons

CSP Word of the Day: Certificate Authorities
- Certificate authorities issue digital certificates that validate the ownership of encryption keys used in secure communications and are based on a trust model.
- Related terms: Public Key Encryption, Cryptography

CSP Word of the Day: Multi-factor Authentication
- Have them search to find out what the latest tech in Multi-factor Authentication
- Related terms: malware,

CSP Word of the Day: Geolocation
- Related terms: Personal Identifiable Information, search history, cookies

Complete each other's Big Idea 4 assignments
- If you completed the 'Create an Assignment' activity from the last class

7.6.2 Khan Academy Review Course
- https://www.khanacademy.org/computing/ap-computer-science-principles
- We used 3 units of this course in our previous activities but there are lots of other review quizzes and readings available. Recommended Units include: Simulations, Data Analysis, Exam Preparation (Practice passage-based questions)

Hot tip from a former student: Khan Academy's Vocabulary list is an excellent last minute study guide. You can find it in the 'Exam Preparation' unit of the Khan Academy course.

7.6.3 Protecting Personal Data Review

- Review outcomes IOC-2.A.1 through IOC-2.B.11
- If you have not had students complete Activity 2 from the AP Teachers Resource (available only through the AP website) or an equivalent activity on protecting Personal Data you should look at it now.

7.6.4 Review Effects of Computing Innovations

- Review the AP CSP Essential Knowledge Statements
    - IOC-1.A.1: People create computing innovations.
    - IOC-1.A.2: The way people complete tasks often changes to incorporate new computing innovations.
    - IOC-1.A.3: Not every effect of a computing innovation is anticipated in advance.
    - IOC-1.A.4: A single effect can be viewed as both beneficial and harmful by different people, or even by the same person.
    - IOC-1.A.5: Advances in computing have generated and increased creativity in other fields, such as medicine, engineering, communications, and the arts.
    - IOC-1.B.1: Computing innovations can be used in ways that their creators had not originally intended:
    - ● The World Wide Web was originally intended only for rapid and easy exchange of information within the scientific community.
    - ● Targeted advertising is used to help businesses, but it can be misused at both individual and aggregate levels.
    - ● Machine learning and data mining have enabled innovation in medicine, business, and science, but information discovered in this way has also been used to discriminate against groups of individuals.
    - IOC-1.B.2: Some of the ways computing innovations can be used may have a harmful impact on society, economy, or culture.
    - IOC-1.B.3: Responsible programmers try to consider the unintended ways their computing innovations can be used and the potential beneficial and harmful effects of these new uses.
    - IOC-1.B.4: It is not possible for a programmer to consider all the ways a computing innovation can be used.
    - IOC-1.B.5: Computing innovations have often had unintended beneficial effects by leading to advances in other fields.
    - IOC-1.B.6: Rapid sharing of a program or running a program with a large number of users can result in significant impacts beyond the intended purpose or control of the programmer.
- Have students complete the *Explore Teachers Resource* Biometric Authentication Activity (available only through the AP website)

7.6.5 AP Exam Official Practice

- Create another practice exam from AP Classroom if needed.
- Be sure to provide a copy of the *Exam Reference Sheet* and discuss if needed.
- It would be good to give them a timed test that mirrors the actual exam's time