Dominican International School
Grade 10 CS Syllabus
SY: 2024-25

Grade Level 11/12
1 Year

Teacher Tim Sampson
Email: tsampson@dishs.tp.edu.tw

Grade 10 Computer Science Discoveries, Design and Physical Computing

Course Description

Computer Science Discoveries 10 is the second half of an introductory computer science course focusing on Innovation and Impact. Students in this course should have already taken the first part in grade 9. The two parts of this course take a wide lens on computer science by covering topics such as programming, physical computing, HTML/CSS, and data. Students are empowered to create authentic artifacts and engage with CS as a medium for creativity, communication, problem solving, and fun. This course uses Code.org's CS Discoveries Curriculum, for more details, please see the 2022-23 Curriculum Guide.

Content

The content covered in this year includes:

- Unit 4 - The Design Process
- Unit 6: Physical Computing

Objectives

Upon completion of unit 4, students should be able to:

- See the design process as a form of problem solving that prioritizes the needs of a user.
- Identify user needs and assess how well different designs address them. In particular they know how to develop paper and digital prototypes, gather and respond to feedback about a prototype, and consider ways different user interfaces do or do not affect the usability of their apps.
- Understand other roles in software development, such as product management, marketing, design, and testing, and to use what they have learned as a tool for social impact.

Upon completion of unit 5, students should be able to:

- Describe the importance of data in solving problems and hypothesize how computers can help in this process.
- Analyze different systems used to represent information in a computer and identify the challenges and tradeoffs posed by using them.
- Explain how collections of data are used to solve problems, and how computers help to automate the steps of this process.
- Give Examples of how the data problem solving process can be applied to an area of your choosing.

Upon completion of unit 6, students should be able to:

- Examine the role of hardware platforms in computing and how different sensors can provide more effective input and output than the traditional keyboard, mouse, and monitor.
- Use App Lab and Adafruit's Circuit Playground, to develop programs that utilize the same hardware inputs and outputs that you see in the smart devices, looking at how a simple rough prototype can lead to a finished product.
- Use the Circuit Playground as the basis for an innovation of your own design.

Classroom Practices

The 6 Main Classroom Practices of CS Discoveries:

- Lead Learner
- Pair Programming
- Think-Pair-Share
- Authentic Choice
- Unplugged Activities
- Peer Feedback

Student Practices

Students in CS Discoveries work in a wide array of contexts, but these experiences are tied together by a core set of practices they develop throughout the course

- Problem Solving
- Persistence
- Creativity
- Collaboration
- Communication

LTO's D'TORCH (Truthful, Organized, Reflective, Courageous and Helpful)

In CS classes the categories of the D'TORCH most practiced and assessed are:

- Organized - Students utilize Google Classroom to edit, submit and keep track of their assignments.
- Reflective - Students will regularly write activity reflections in their online journal.
- Helpful - Students are empowered to ask for and provide explanations and give examples to help classmates through particularly difficult problems.

Class Expectations

- Come to class on time and be prepared
- Have a positive attitude and be willing to learn.
- Respect yourself, others, and our school.
- Always complete your work and try your best.
- Actively participate, listen carefully, but don't speak out of turn.
- All assignments must be completed.

Homework and Quiz Rules

- All assignments must be turned in on the day they are due.
- 1 day late = Maximum of only 60%
- 2+ days late = Project-I & Only 60%
- If a student has been absent, it is his/her duty to find out what work is due, and hand it in a day later.
- All assignments must satisfactorily be completed.
- If you are absent on the day of the quiz, you will only be able to get a maximum of 60%.

Classroom Rules

- All students are expected to follow the rules. Consequences will follow if rules are broken.
- Read and follow the standard school rules.
- Be on time and neatly dressed, in full school uniform.
- Speak in ENGLISH ONLY.
- Respect your teachers, fellow students and their property.
- Keep your seating space and classroom clean and neat.
- No eating or drinking in the ICT Labs.
- Ask permission to leave the class.

Academic Dishonesty means employing a method or technique or engaging in conduct in an academic endeavor that contravenes the standards of ethical integrity expected at DIS. Academic dishonesty includes but is not limited to, the following:

1. Purposely incorporating the ideas, words of sentences, paragraphs, or parts thereof without appropriate acknowledgment and representing the product as one's own work; and
1. Representing another's intellectual work such as photographs, paintings, drawings, sculpture, or research or the like as one's own, including failure to attribute content to an AI.
2. Employing a tutor, making use of Artificial Intelligence without acknowledgement, getting a parent to write a paper or do an assignment, paying for an essay to be written by someone else and presented as the student's own work.
3. Committing any act that a reasonable person would conclude, when informed of the evidence, to be a dishonest means of obtaining or attempting to obtain credit for academic work.

Any act of academic dishonesty will result in an automatic zero on the entire assignment

Discipline

- Verbal warning
- Write-Up, entered into the discipline system and then referral to the Discipline Office.
- Parent-Teacher conference as required.

Links, tools and references:

- https://code.org/educate/csd
- https://developer.mozilla.org/en-US/docs/Learn
- https://www.w3schools.com/
- App Lab — A browser-based JavaScript programming environment for creating interactive apps, with the ability to freely switch between programming in blocks or text
- Maker Toolkit — A collection of commands that extends App Lab's capabilities to allow students to easily program the Circuit Playground and many other physical computing devices directly from App Lab
- Circuit Playground — Adafruit's new low-cost Arduino-based microcontroller featuring multiple integrated sensors and output devices

Schedule for Computer Science Discoveries, Design, Data and Physical Computing

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)	
Week / Date	Topic / Projects / Assessments

Week 1 **Aug 12th to 16th** **4 Days of Class** 12~ First Day / Orientation Day 15~ Opening Mass & Assumption of Our Lady 8:00 15~ Induction of Class, Student Council Officers and DYM	**The Design Process** The Design Process unit transitions students from thinking about computer science as a tool to solve their own problems towards considering the broader social impacts of computing. Through a series of design challenges, students are asked to consider and understand the needs of others while developing a solution to a problem. The second half of the unit consists of an iterative team project, during which students have the opportunity to identify a need that they care about, prototype solutions both on paper and in App Lab, and test their solutions with real users to get feedback and drive further iteration. **Lesson 1: Designing with Empathy** The class explores a variety of different shoe designs to consider design choices. Building on this, students explore the relationship between users, their needs, and the design of objects they use.
Week 2 **Aug 19th to 23rd**	**Lesson 2: Understanding Your User** Using user profiles, students explore how different users might react to a variety of products. Role playing as a different person, each member of the class will get to experience designs through someone else's eyes. **Lesson 3: User-Centered Design - Define and Prepare** In small groups, students use the design process to come up with ideas for smart clothing. Today's lesson focuses on brainstorming users and ideas that will meet their needs. Over the course of both lessons, students will brainstorm ideas, identify users, and finally propose a design. This activity serves as the first of several opportunities for students to practice designing a solution for the needs of others.
Week 3 **Aug 26st to 30th** 26~Fire drill? 26~Middle and High School Catholic Bridge Program (after assembly) 28~St. Dominic de Guzman Feast Day Celebration	**Lesson 4: User-Centered Design - Try and Reflect** In small groups, students will use the design process to come up with ideas for smart clothing. Today's lesson focuses on creating a design and reflecting on how well it meets the needs of users. Over the course of both lessons, students will brainstorm ideas, identify users, and finally propose a design. This activity serves as the first of several opportunities for students to practice designing a solution for the needs of others. **Lesson 5: User Interfaces** In this lesson, students get to see how a paper prototype can be used to test and get feedback before writing any code. To help out a developer with their idea, the class tests and provides feedback on an app prototype made of paper.
Week 4 **Sep 2nd to 6th** 2~House Ceremony	**Lesson 6: Feedback and Testing** Users have been testing an app, and they have lots of feedback for the developer. The class needs to sort through all of this feedback, identify the common themes and needs, and start revising the prototype to make it better meet the users' needs. **Lesson 7: Identifying User Needs** In this lesson, the class begins thinking about designing their own paper prototype for an app that can solve a problem in our community. Using interviews from different users, students identify needs and interests that they can use to design an app for these people in their community.
Week 5 **Sep 9th to 13th** 9~ Mass & Birthday Mother Mary& VIP Induction	**Lesson 8: Project - Paper Prototype** Using the interview information from the previous lesson, students come up with app ideas to address the needs of their users. To express those ideas, and test out their effectiveness, students create and test paper prototypes.
Week 6 **Sep 16th to 20th** **1 Day of Class** 17~Moon Festival 18-20~ Teacher's Conference	**Lesson 9: Designing Apps for Good** To kick off the app design project, the class organizes into teams and starts exploring app topics. Several examples of socially impactful apps serve as inspiration for the project. they are working on, and use those apps to help refine the app idea they will pursue.

Week 7 **Sep 23rd to 27th** 24-26~Pre-Exam Days	**Lesson 10: Market Research** In this lesson, the class dives into app development by exploring existing apps that may serve similar users. In groups, students will identify a handful of apps that address the same topic **Lesson 11: Exploring UI Elements** Paper prototypes allow developers to quickly test ideas before investing a lot of time writing code. In this lesson, teams explore some example apps created in App Lab and use these examples to help inform the first paper prototypes of their apps.
Week 8 **Sep 30th to Oct 4th**	**Q1 Final Exam**
Week 9 **Oct 7th to 11th** **1 Day of Class** 7~Launching - Rosary Month and Bullying Prevention Day 8-9 ~Q1 Exams 10~Double Ten 11~Record Day	**Lesson 12: Build a Paper Prototype** In teams, students will create a paper prototype for the app they've been developing. Each team member will create a different screen and design how the user will navigate between each screen.

2nd QUARTER – TENTATIVE COURSE CONTENT

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)

Week / Date	Topic / Projects / Assessments
Week 1 (10) **Oct 14th to 18th** 14~ Second Quarter Begins	**Lesson 13: Prototype Testing** In this lesson, teams test out their paper prototypes with other members of the class. As one student role plays as the computer, one narrates, and the rest observe, teams will get immediate feedback on their app designs, which will inform the next version of their app prototypes. **Lesson 14: Design Mode in App Lab** Teams now move to App Lab to build the next iteration of their apps. This lesson focuses on how to use Design Mode in App Lab to create digital prototypes for their apps.
Week 2 (11) **Oct 21st to 25th** 25 – Book Fair 25- Masquerade Night	**Lesson 15: Build a Digital Prototype** Using the drag-and-drop Design Mode, each team member builds out at least one page of their team's app, responding to the feedback received in the previous round of testing. **Lesson 16: Events in App Lab** Building on the previous lesson, we learn how to import new screens into our apps and link them together using buttons and events to complete the Recycle Finder app we started in an earlier lesson.
Week 3 (12) **Oct 28th to Nov 1st** 1-All Saint's Day Mass	**Lesson 17: Linking Prototype Screens** Building on the screens that they designed in the previous lesson, teams combine screens into a single app. Simple code can then be added to make button clicks change to the appropriate screen. **Lesson 18: Testing the App** In this lesson, teams run another round of user testing with their interactive prototype. Feedback gathered from this round of testing will inform the final iteration of the digital prototype.
Week 4 (13) **Nov 4th to Nov 8th**	**Lesson 19: Bugs and Features**

	Teams analyze the feedback they received from the last round of testing and make a plan for how they would like to address it. Students categorize feedback as either a bug or a feature and decide which items are most important for improving their app. **Lesson 20: Updating Your Prototype** Using the feedback from the last round of testing, teams implement changes that address the needs of their users. Each team tracks and prioritizes the features they want to add and the bugs they need to fix.
Week 5 (14) **Nov 11th to 15th**	**Lesson 21: Project - App Presentation** Each team prepares a presentation to "pitch" the app they've developed. This is the time they can share the struggles, triumphs, and plans for the future.
Week 6 (15) **Nov 18th to 22nd** 22-Gr.12 Q2 Exam 22 - YSC Contest	**Physical Computing** **Lesson 1: Intro to App Lab** This tutorial is designed to quickly introduce the App Lab programming environment as a powerful tool for building and sharing apps. The tutorial itself teaches students to create and control buttons, text, images, sounds, and screens in JavaScript using either blocks or text. At the end of the tutorial, students are given time to either extend a project they started building into a "Choose Your Own Adventure", "Greeting Card", or "Personality Quiz" app. They can also continue on to build more projects featured on the code.org/applab page. can gradually start to integrate elements of the board as an output device while relying on App Lab for user input.
Week 7 16) **Nov 25th to 29th** 25-Gr.12 Q2 Exam 26-28~Pre-Exam Day	**Lesson 2: Physical Designs** To kick off a unit devoted to problem-solving and creating apps with devices, students begin by investigating the design of different physical devices and their apps. Students look at a variety of physical designs and attempt to match each design with a potential user. Then students choose a user and attempt to prototype a physical design for them on paper or in a digital template. To conclude the activity, students consider what it means to be a physical designer and create resources for other users. **Lesson 4: Board Events** This lesson transitions students from considering the Circuit Playground as strictly an output device and instead introduces the buttons and toggle switches as tools for input. Starting with the hardware buttons and switch, students learn to use onBoardEvent(), analogously to onEvent(), in order to take input from their Circuit Playgrounds.
Week 8 (17) **Dec 2nd to Dec 6th** **6~Half Day** Foundation Day Celebrations	**Lesson 5: Board Events** Using the hardware buttons and switch, students develop programs that use the Circuit Playground as an input. **Lesson 6: Variables and If Statements** In this lesson, students are introduced to variables, the counter pattern, and if-statements. Students will use these concepts in the context of programming the circuit playground and creating more complex input/output behaviors, such as counting the number of button presses before having the circuit playground make a noise. The concepts in this lesson are used to make more complex and fulfilling apps in future lessons, and students may need to refer back to the videos and examples in this lesson as they continue to master these concepts.
Week 9 (18) **Dec 9th to 13th** **3 Days of Class** 12-13 ~Q2 Exams	**Q2 Final Exam**
Dec 16th to Jan 3rd	**Christmas Break**

3rd QUARTER – TENTATIVE COURSE CONTENT

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)

Week / Date	Topic / Projects / Assessments
Week 1 (19) **Jan 6th to 10th** **4 Days of Class** 6~Record Day 7~Third Quarter Begins 10 ~ New Year Mass	**Lesson 7: Mini-Project - Field Collector App** In this mini-project, students will create an app that uses the Circuit Playground to collect data, then has an app to analyze the data that was collected. This is similar to citizen science fieldwork or survey apps that students may be familiar with. Students will use variables and events to collect data from the circuit playground, then use if-statements to make decisions or recommendations based on the data they collect.
Week 2 (20) **Jan 13th to 17th**	**Lesson 8: Color LEDs** In this lesson, students learn how to use the 10 color LEDs on the Circuit Playground. Students will control the color and intensity of each LED, then use what they have learned to program light patterns to create a light show on their Circuit Playground. **Lesson 9: Getting Screen Inputs** Students learn to use several new design elements - text inputs, dropdowns, and sliders - so they can get user input from the screen of their apps. This lesson also introduces the getProperty and getText blocks, which allow them to access their user input in their code. Students later use getProperty and setProperty together with the counter pattern to make elements move across the screen. A new event trigger, change, is also introduced to represent when a dropdown or slider changes values.
Week 3 (21) **Jan 20th to 24th**	**Lesson 10: Project: Human Device Interaction** In this project, students create an app that controls the Circuit Playground so it interacts with the physical environment around it, similar to many smartphone apps that are used to control devices in a house or car or school. Students use physical materials to help design their physical device, then create an app that lets the user interact with the physical device or change settings. o directly interact with it at all or may interact without actually realizing they are doing so.
Jan 27th to Jan 31st	**CNY Holiday**
Week 4 (22) **Feb 3rd to 7th**	**Lesson 11: Board Sensors** In this lesson, students explore how the three sensors (sound, light, and temperature) can be used to write programs that respond to changes in the environment. This marks a transition in terms of how users interact with a program. By using sensors as an input, the user of an app doesn't have t
Week 5 (23) **Feb 10th to 14th** 1-14~Catholic Week	**Lesson 12: Accelerometer** In this lesson, students will explore the accelerometer and its capabilities. They'll become familiar with its events and properties, as well as create multiple programs utilizing the accelerometer similar to those they've likely come across in real world applications.
Week 6 (24) **Feb 17th to 21st**	**Lesson 13: Making Music** In this lesson students will use the buzzer to its full extent by producing sounds, notes, and songs with the buzzer. Students start with a short review of the buzzer's frequency and duration parameters, then move on to the concept of notes. Once students are able to play notes on the buzzer, they use arrays to hold and play sequences of notes, forming simple songs.
Week 7 (25) **Feb 24th to 28th** **4 Days of Class** 24~Lenten Mass?	**Lesson 14: Functions** This lesson introduces students to functions as a way to organize and group repeated blocks of code together, such as changing all of the LEDs to red or blue. Then, students learn how to use

25-27 ~ Pre-Exam Days 24-27~IOWA Assessments 28 ~ Memorial Day Holiday	parameters in their functions as a way to generalize behaviors to work for different contexts - for example, changing all the LEDs to a certain color rather than always changing them red or blue.
Week 8 (26) **March 3rd to 7th** 5~ Ash Wednesday	**Q3 Final Exam**
Week 9 (27) **March 10th to 14th** **4 Days of Class** 14 – Q3 Exams	**Lesson 15: Mini-Project - Interactive Art** In this lesson, students create a piece of interactive artwork using the sensors on the Circuit Playground and physical materials. Students explore how the Circuit Playground can augment physical materials to create an interactive experience. This project does not use the App Lab screen except to help with debugging - otherwise, users only interact with the physical artwork itself.

4th QUARTER – TENTATIVE COURSE CONTENT

(NB: Depending on time and interest, the teacher may delete and/or add other selections.)	
Week / Date	Topic / Projects / Assessments
Week 1 (28) **March 17th 21st** **4 Days of Class** 17 – Q3 Exams 18~ Fourth Quarter Begins 18~ Fire Drill? 19~ Feast of St. Joseph	**Lesson 15: Mini-Project - Interactive Art** In this lesson, students create a piece of interactive artwork using the sensors on the Circuit Playground and physical materials. Students explore how the Circuit Playground can augment physical materials to create an interactive experience. This project does not use the App Lab screen except to help with debugging - otherwise, users only interact with the physical artwork itself.
Week 2 (29) **March 24th to 28th**	**Lesson 16: Physical Outputs and LEDs** In this lesson, students learn how to attach external LEDs to their circuit playground and use code to light up these LEDs. This allows students to create more flexible devices that mimic real-world products. This lesson requires several external materials, as well as safety protocols to ensure students don't accidentally damage their circuit playground.
Week 3 (30) **March 31st to April 4th** **4 Days of Class** 4~Tomb Sweeping	Lesson 17: Physical Outputs and LEDs In this lesson, students learn how to attach external LEDs to their circuit playground and use code to light up these LEDs. This allows students to create more flexible devices that mimic real-world products. This lesson requires several external materials, as well as safety protocols to ensure students don't accidentally damage their circuit playground.
Week 4 (31) **Apr 7th to 11t**	Lesson 18: Physical Inputs and Buttons In this lesson, students learn how to connect external wires to create input events when the wires touch, simulating a button press. Students learn to use code to recognize these external button events to make changes to their app. This allows students to create more flexible devices that mimic real-world products. This lesson requires several external materials, as well as safety protocols to ensure students don't accidentally damage their circuit playground.
April 14th to April 18th	**Easter Break**
Week 5 (32) **Apr 21st to 25th**	**Lesson 18: Physical Inputs and Buttons** In this lesson, students learn how to connect external wires to create input events when the wires touch, simulating a button press. Students learn to use code to recognize these external button events to make changes to their app. This allows students to create more flexible

	devices that mimic real-world products. This lesson requires several external materials, as well as safety protocols to ensure students don't accidentally damage their circuit playground.
Week 6 (33) **Apr 28th to May 2nd**	**Lesson 19: Project - Prototype an Innovation** In this final project for the course, students team to develop and test a prototype for an innovative computing device based on the Circuit Playground. Using the inputs and outputs available on the board, groups will create programs that allow for interesting and unique user interactions.
Week 7 (34) **May 5th to 9th**	**Lesson 19: Project - Prototype an Innovation** In this final project for the course, students team to develop and test a prototype for an innovative computing device based on the Circuit Playground. Using the inputs and outputs available on the board, groups will create programs that allow for interesting and unique user interactions.
Week 8 (35) **May 12th to 16th**	**Q4 Final Exam**
Week 9 (36) **May 19th to 23rd**	19-23 ~ Student Clearance 19~ Baccalaureate Mass 23~Gr. 6 – 7 Recognition and Gr. 8 Graduation
Week 10 (37) **May 26th to 30th**	4 Days of Class 26~House Culminating Activity 27~Gr. 9-11 Recognition and Gr. 12 Graduation 28! Class Party 29- ~ Students Last Day 30~ Teachers/Staff Meeting

The end ~ Have a great summer 😊

CS Subject Sequence 24-25

<table>
<tr><th colspan="5">High School CS Curriculum</th></tr>
<tr><th>Type</th><th>Classes (45m)</th><th>HW (45m)</th><th colspan="2">Grade, Curriculum and Description</th></tr>
<tr><td rowspan="3">Subject CS</td><td rowspan="3">2</td><td rowspan="3">2</td><td><b>G09 CS Discoveries</b></td><td><b>G10 CS Discoveries</b></td></tr>
<tr><td>Code.org Discoveries
Unit 1 Problem Solving and Computing
Unit 2 Web Development
Unit 3 Animations and Games</td><td>Unit 4 - The Design Process
Unit 6: Physical Computing</td></tr>
<tr><td><b>G11 CS Principles</b></td><td><b>G12 CS Principles</b></td></tr>
</table>

			Code.org CS Principles Unit 1 - Digital Information Unit 2: The Internet Unit 9: Data Unit 10: Cybersecurity and Global Impacts Unit 7: AI and Machine Learning (CSD)	Unit 5 Building Apps Unit 4 Big Data and Privacy Unit 6 Making Data-backed Apps
AP	6	6	**G11 APCS A JAVA** **CSAwesome**	**G12 APCS Principles** **CS50AP**
			The course introduces students to computer science with fundamental topics that include problem solving, design strategies and methodologies, organization of data (data structures), approaches to processing data (algorithms), analysis of potential solutions, and the ethical and social implications of computing.	This course offers a multidisciplinary approach to teaching the underlying principles of computation. The course introduces students to computer science with fundamental topics that include problem solving, design strategies and methodologies, organization of data (data structures), approaches to processing data (algorithms), analysis of potential solutions, and the ethical and social implications of computing.

High School CS Curriculum Overview

Our computer science curriculum is designed to provide a comprehensive and flexible learning experience from grades 9 through 12, catering to both potential CS majors and students seeking a well-rounded CS education.

Curriculum Progression and Options

1. **Grades 9-10: CS Discoveries**

   - Foundational for all students
   - Covers problem-solving, web development, animations, games, and the design process
   - Introduces physical computing concepts

2. **Grades 11-12: Flexible Pathways**

   a) **Minor Subject Track: CS Principles**

   - Ideal for non-CS majors or those seeking a science AP credit
   - Builds on CS Discoveries with more advanced topics
   - Explores digital information, the Internet, data analysis, cybersecurity, and machine learning
   - Provides a well-rounded CS experience without the intensity of the AP track

   b) **AP Track for Prospective CS Majors**

   - Grade 11: APCS A JAVA
     - Introduces fundamental CS topics with a focus on Java programming
     - Covers problem-solving, design strategies, data organization, and algorithmic approaches
   - Grade 12: CS50AP (AP Computer Science Principles)
     - Culminating course offering a multidisciplinary approach to computation

    - Prepares students for college-level CS and the AP exam

Curriculum Flexibility and Benefits

1. **Options for Various Academic Paths**:

   - Students not planning to major in CS can take CS Principles in grades 11 and 12 as a minor subject, fulfilling science AP credit requirements while gaining valuable CS knowledge.
   - Those considering a CS major in college can opt for the more intensive AP track.

2. **Well-Rounded CS Experience**:

   - The CS Principles track ensures students gain a comprehensive understanding of CS concepts without the rigorous demands of AP courses.
   - Ideal for students interested in CS as a complementary skill to their primary academic focus.

3. **Preparation for CS Majors**:

   - The AP track provides in-depth preparation for students planning to pursue CS in college.
   - APCS A JAVA and CS50AP offer college-level content and prepare students for advanced studies.

4. **Flexibility to Change Paths**:

   - Students can reassess their interests and switch tracks between grades 10 and 11 if their academic goals change.

CS50AP as the Capstone for AP Track

For students on the AP track, CS50AP serves as a rigorous capstone, building upon APCS A JAVA and previous coursework. Its comprehensive nature makes it an ideal final course, covering advanced topics and preparing students for college-level CS studies.

Practical Application

To complement both curriculum tracks, we encourage all CS students to apply their skills through our Service Learning program. The HS CS department collaborates with this program to help students identify opportunities where they can use their computer science knowledge in real-world contexts, enhancing their learning experience regardless of their chosen track.

Curriculum Development and Stakeholder Feedback

At our school, we are committed to continuously evaluating and improving our CS curriculum to ensure it meets the needs of our students and prepares them for future academic and career challenges. Our approach includes:

1. **Curriculum Trials and Evaluation**:

   - We regularly explore potential additions to our curriculum. For example, in previous years, we conducted trials of CS50 SQL and CMU's College Level Programming courses.
   - These trials helped us assess the value and fit of new courses within our existing framework.

2. **Rigorous Assessment**:

   - Through these trials, we found that even with highly capable and enthusiastic students, our current AP track, culminating in CS50AP, already provides sufficient content, topics, and rigor.
   - This reinforced our confidence in the comprehensive nature of our existing curriculum.

3. **Stakeholder Engagement**:
    - We actively seek and encourage feedback from all stakeholders, including students, parents, administrators, and industry professionals.
    - This collaborative approach ensures our curriculum remains relevant and aligned with both academic standards and real-world needs.

4. **Adaptive Planning**:
    - Based on stakeholder input, we continually refine our approach to practical skill application.
    - For instance, after extensive consultation, we determined that integrating industry-related skills and community engagement through our existing Service Learning program was the most effective approach.

5. **Ongoing Collaboration**:
    - The High School CS department works closely with the Service Learning program to help students identify opportunities to apply their CS skills in meaningful ways.

Our commitment to curriculum development and stakeholder feedback ensures that our CS program remains dynamic, relevant, and responsive to the evolving needs of our students and the broader community.

Practical Application through Service Learning

Building on our stakeholder feedback, we are focusing future efforts towards encouraging students to make use of our existing Service Learning program. This approach allows students to:

- Apply their CS skills in real-world contexts within the community
- Gain valuable experience that complements their classroom learning
- Develop a deeper understanding of how CS can be used to address real-world challenges

As this initiative evolves, the HS CS department continues to work closely with the Service Learning program to identify and create opportunities that allow students to maximize the practical application of their CS skills.