Dominican International School

Grade 10 CS Syllabus

SY: 2025-26

Grade Level 10
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 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://www.w3schools.com/
- 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

1st QUARTER – TENTATIVE COURSE CONTENT

Week/Date	Topic/Projects/Assessments
Week 1 (Aug 12 to 15)	**Monday** No School **The Design Process** **Chapter 1: User Centered Design** Lesson 1: Designing With Empathy
Week 2 (Aug 18 to 22)	Lesson 2: Understanding Your User Lesson 3: User-Centered Design - Define and Prepare
Week 3 (Aug 25 to 29)	**Q1 Quiz 1** Lesson 4: User-Centered Design - Try and Reflect
Week 4 (Sept 1 to 5)	Lesson 5: User Interfaces Lesson 6: Feedback and Testing
Week 5 (Sept 8 to 12)	Lesson 7: Identifying User Needs Lesson 8: Project - Paper Prototype
Week 6 (Sept 15 to 19)	**Q1 Quiz 2** **Chapter 2: App Prototyping** Lesson 9: Designing Apps for Good
Week 7 (Sept 22 to 26)	Lesson 10: Market Research **Q1 Final Exam**
Week 8 (Sept 29 to Oct 3)	**Monday** No School - l Confucius' Birthday **Tuesday** Regular Schedule **Wednesday** Major Exams **Thursday** Major exams **Friday** No Students Record Day
Week 9 (Oct 6 to Oct 10)	**Monday** No School Oct 6th Moon Festival, **Tuesday - Thursday** No Class Teachers Conference **Friday** No School October 10th Double10 day

2nd QUARTER – TENTATIVE COURSE CONTENT

Week/Date	Topic/Projects/Assessments
Week 1 (10) (Oct 13 to 17)	**Monday** No School Record Day Lesson 11: Exploring UI Elements
Week 2 (11) (Oct 20 to 24)	Lesson 12: Build a Paper Prototype Lesson 13: Prototype Testing
Week 3 (12) (Oct 27 to 31)	**Q2 Quiz 1** Lesson 14: Design Mode in App Lab
Week 4 (13) (Nov 3 to 7)	Lesson 15: Build a Digital Prototype Lesson 16: Events in App Lab
Week 5 (14) (Nov 10 to 14)	Lesson 17: Linking Prototype Screens Lesson 18: Testing the App
Week 6 (15) (Nov 17 to 21)	**Q2 Quiz 2** Lesson 19: Bugs and Features
Week 7 (16) (Nov 24 to 28)	Lesson 20: Updating Your Prototype Lesson 21: Project - App Presentation P1
Week 8 (17) (Dec 1 to 5)	**Q2 Final Exam** Lesson 21: Project - App Presentation P2
Week 9 (18) (Dec 8 to 12)	**Monday** No Classes - Foundation Day Mass, cake ceremony, and Class Party (half day) **Thursday** Major Final Exams half day **Friday** Major Final Exams half day
December 15 to January 2 Christmas Break	

Third Quarter Tentative Course Content

Week/Date	Topic/Projects/Assessments
Week 1 (19) (Jan 5 to 9)	**Monday** No Students Record Day **Unit 6 Physical Computing** Lesson 0: Intro to App Lab

Week 2 (20) (Jan 12 to 16)	Lesson 1: Physical Designs Lesson 2: The Circuit Playground Lesson 3: Updating Screen Elements Lesson 4: Board Events
Week 3 (21) (Jan 19 to 23)	**Q3 Quiz 1**
Week 4 (22) (Jan 26 to 30)	Lesson 5: Variables and If Statements Lesson 6: Mini-Project - Field Collector App
Week 5 (23) (Feb 2 to 6)	Lesson 7: Color LEDs Lesson 8: Getting Screen Inputs Lesson 9: Combining Inputs and Outputs
Week 6 (24) (Feb 9 to 13)	**Q3 Quiz 2** Lesson 10: Project: Human Device Interaction
February 16-20 Chinese New Year Holiday	
Week 7 (25) (Feb 23 to 26)	Lesson 11: Board Sensors **Friday** Memorial Day Holiday (no classes)
Week 8 (26) (Mar 2 to 6)	Lesson 12: Accelerometer **Q3 Final Exam**
Week 9 (27) (Mar 9 to 13)	Lesson 13: Making Music **Friday** Third Quarter Exam (half day)

Fourth Quarter Tentative Course Content

Week/Date	Topic/Projects/Assessments
Week 1 (28) (Mar 16 to 20)	**Monday** Third Quarter Exam (half day) Lesson 14: Functions
Week 2 (29) (Mar 23 to 27)	Lesson 15: Mini-Project - Interactive Art Lesson 16: Physical Outputs and LEDs
Mar 30 to Apr 6 Easter / Spring Break	

Week 3 (30) (Apr 7 to 10)	**Monday** No School Spring Break **Q4 Quiz 1**
Week 4 (31) (Apr 13 to 17)	Lesson 17: Physical Outputs and LEDs Lesson 18: Physical Inputs and Buttons
Week 5 (32) (Apr 20 to 24) 20-24 AP Mock Exams	**Q4 Quiz 2** Lesson 19: Project - Prototype an Innovation - Part 1: Define - Scope Innovation
Week 6 (33) (Apr 27 to 30) 27-30 Senior Project Presentations 28-30 Pre-Exam Days	Lesson 19: Project - Prototype an Innovation - Part 2 Prepare - Complete Project Guide **Q4 Final Exam**
Week 7 (34) (May 4 to 8) 4-14 Final Exams (K, Gr. 5, 8, & 12 Only) 4-15 AP Exams	**Monday** No School Labor Day **Lesson 19: Project - Prototype an Innovation Part 3 - 5** - Try - Develop Prototypes - Reflect - Peer Review - Iterate - Revise Prototypes
Week 8 (35) (May 11 to 15)	**Wednesday** Major Exams Day 1 Half Day **Thursday** Major Exams Day 2 Half Day **Friday** No Students Record Day 1
Week 9 (35) (May 18 to 22)	Special Events
Week 10 (36) (May 25 to 29)	Special Events

The end ~ Have a great summer 😊

CS Subject Sequence 25-26

The table below outlines the available high school computer science (HS CS) pathways at DIST. Each class is 45 minutes long, and most students are able to complete lessons within that time. Students who need extra practice or review may need an additional 15 minutes to show understanding.

Students who wish to achieve deeper mastery may choose to spend more time on lessons. Enrichment opportunities are always available within the lessons, and in some cases additional activities are provided. These enrichment activities are not part of the course grade but give students valuable chances for extra practice and self-assessment.

Ultimately, a student's effort directly influences their individual growth and success in computer science. While enrichment work does not guarantee higher grades, it helps build stronger skills and deeper understanding.

High School CS Curriculum	
Grade, Curriculum and Description	
G09 CS Discoveries	**G10 CS Discoveries**
Code.org Discoveries Unit 1 Problem Solving and Computing Unit 3 Animations and Games	Code.org Discoveries Unit 4 - The Design Process Unit 6: Physical Computing
G11 CS Principles	**G12 CS Principles**
Code.org CS Principles Unit 1 - Digital Information Unit 5: Data Unit 7 (CSD): AI and Machine Learning	Code.org CS Principles Unit 3 - Intro to App Design Unit 4 - Variables, Conditionals, and Functions Unit - 6 Lists, Loops, and Traversals Unit - 7 Parameters, Return, and Libraries
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.