Computer Science Discoveries (CSD)

Course Overview: Computer Science Discoveries introduces you to a variety of computer science skills and projects. We'll solve problems like computers; we'll design websites; we'll make games; we'll build apps; we'll analyze data; we'll control electronics – and we'll make lots of projects along the way. This is a true introductory course – **you don't have to know anything about Computer Science in order to take this class**. By the end of the course, we will have experienced several aspects of being a computer scientist, hopefully having fun along the way and maybe you might even decide you'd like to take more Computer Science classes here at Amphi (like AP Computer Science Principles).

This is **not a Math class**. This is **not a Programming-Only class**. This is **not a Sit-and-Listen class**.
This is a *make something* class. This is a *debate something* class. This is a *present something* class.

Unit 1: Think Like a Computer	→	Our first unit focuses on the differences between how a computer solves problems and how a human solves problems. We look at examples of modern technology and the types of problems they were built to solve, then try to train ourselves to 'think like a computer' to solve our own problems. A variety of activities and problems are introduced in this unit to help develop our ability to give clear instructions to a computer.
Unit 2: Web Design	→	Now that we can think like a computer, we start building websites using HTML and CSS. Basic commands are introduced to put content on a webpage, add decoration and style to our content, and link our webpages together. We'll create a personal website that will become our portfolio for the rest of our projects throughout the year.
Unit 3: Game Design	→	Next we transition to making games using the Code.org Game Lab. Using a version of Javascript, we'll create sprite-based games like Mario or Tetris or other classic games. Event-driven programming is emphasized along with the concepts of events, a 'game loop', and re-drawing the screen for animations. Basic sprite techniques and programming logic are also introduced. This unit culminates in a 'create your own game' project.
Unit 4: App Design	→	In this unit, we shift our motivation from making programs for *ourselves* to making programs for *someone else*. Using the Code.org App Lab, we start to investigate how to create multi-screen apps that react to user input. User-interface design and events are introduced in this unit, with an emphasis on designing for a touch-screen phone rather than a website. Throughout the unit, we will find a partner in the community to partner with to create an app that suits the need of this person. This mimics the developer-client relationship that software engineers experience, where you design products *for a client* rather than just for your own personal motives. This unit culminates in a presentation of your app, it's user interface, and how you worked with your client to create the final design and product.
Unit 5: Computers & Data	→	We take a closer look at all of the projects and creations we've made this year and think about how a computer actually *represents* and *collects* the data we've used in our projects. We learn how computers represent numbers, letters, and images using the building blocks of all computers: binary numbers. We then look at the big picture and how computers can collect and analyze data, especially our personal data that might be floating around the world. Issues of privacy and security are discussed, as well as ways to be aware of our digital footprint.
Unit 6: Electronics	→	In this final unit, we look at how tiny electronic computers (like your calculator) interact with the world and how we can make specialized programs just for them. We use a device called a Circuit Playground to create electronics that interact with our world using sensors, lights, and sounds. We spend time programming these Circuit Playgrounds using the Code.org Maker Lab, working towards a project of your choice that uses the sensors from the Circuit Playground to control an app on your computer. During this unit, we also look at the prevalence of electronics and computers in the world around us (like in our car and refrigerator) and discuss how widespread technology has become.

CSD Big Ideas

Problem Solving • Web Development • Games & Animations • App Design • Data & Society • Physical Computing

How To Participate In This Class

This class will feature a lot of *projects*. This class will feature a lot of *collaboration*. This class will feature a lot of *explaining to your neighbor*. As such, there are certain behaviors that will help you be successful in this class. At the very least, you are expected to:

Manage your time so you can finish assignments and be prepared for class discussions	Read the board, check your email, and communicate with your peers so you are prepared for class	Contribute positively when working in groups or with a partner
Speak up when you don't understand something – you're probably not the only person with a question	Work independently without getting distracted	Prepare for tests and quizzes by reviewing and studying your notes and assignments
Keep your Promises	**Be In Class**	**Complete Your Work**

How You Earn Your Grade

Projects & Papers: This category is worth **30%** of your grade. These are larger assignments that might take a few days to complete because they come at the end of a unit where you need to put *everything* together to complete a task. They usually involve some sort of creativity and time-management to make sure you finish on time.

Programs & Homework: This category is worth **25%** of your grade. These are infrequent assignments that test your knowledge of the concepts we've covered so far. They are typically due a week or so after they are assigned so you have a chance to work on them early and ask questions. You can even submit them early and I will give you feedback so you can edit and re-submit. You can turn in these assignments late, the highest grade you can get on a late assignment is a B. The only way to earn an A on assignments if for them to be correct *and* on time.

Quizzes & Tests: This category is worth **20%** of your grade. These are usually in-class quizzes or shorter tests to see what you know individually and without any help. Most Computer Scientists collaborate to solve problems and continually work to revise and fix their mistakes, so we won't always have a ton of tests in this class. This is also the category worth the *least* number of points – your best shot at getting a good grade in this class is turning in your projects and homework on time rather than by passing the tests.

Everything Else: This category is worth **25%** of your grade. It's everything else that doesn't fit in the category above, but it's usually activities we complete in class or with a group. For this category, it's especially important that you're in class to earn these points and be successful.

Class Fee

This class has a fee of $11. This fee is used to purchase supplies for the Electronics portion of the class in the second semester. Any supplies purchased in this way are for **you to keep**. We are very flexible about *when* and *how* you can pay this fee – please see Schneider if you have any questions or concerns.