

THINGS MISSING IN HIGH SCHOOL COMPUTER SCIENCE

Eshaan Punalekar



INTRODUCTION

Being a high schooler myself on the computer science path, I often find myself searching for different topics in the field that lie past the scope of generic high school courses (AP Computer Science Principles, AP Computer Science Applications, and any other basic courses your high school offers). However, I firmly believe that these high school courses are much needed to grasp the intricacies of complex computer science topics. Consequently, I have decided to create a comprehensive guide that thoroughly explores topics that I believe every high school student looking to leave college with a CS degree should know of. Throughout this guide, I will be providing links to research papers, courses, blogs, and other websites that will provide you with extra information on the topics covered.

Before we start, let's go over the basics. According to an [article](#) by Britannica, computer science is defined as "*the study of computers and computing as well as their theoretical and practical applications. Computer science applies the principles of mathematics, engineering, and logic to a plethora of functions, including algorithm formulation, software, and hardware development, and artificial intelligence.*" Average high school computer science courses thoroughly explore computer functionality (parts of a computer etc.), offer a light exposure to ordinary coding languages (Python, Java), and hardly encourage application-level software development. This guide dives into the worlds of artificial intelligence and machine learning, cybersecurity, web development, web design, software development, and app development.

Throughout this guide, you will come across courses and projects that require coding. Knowing Python and Java are a must! If needed, you can check out a full Python tutorial on YouTube [here](#). You can also check out a full Java tutorial on YouTube [here](#). In addition, here are a few IDEs that, in my opinion, are among the best: [PyCharm Community Version](#) (Python), [IntelliJ IDEA](#) (Java), [Sublime Text](#) (HTML, Javascript, other), and [XCode](#) (Mac users - Swift).

So, let's dive right in.

ARTIFICIAL INTELLIGENCE

Artificial intelligence is often linked to humanoids, robots, and the concept of how machines will eventually overpower humans. However, artificial intelligence is nothing more than tons and tons of complex algorithms comprising a bunch of clever math and long equations.

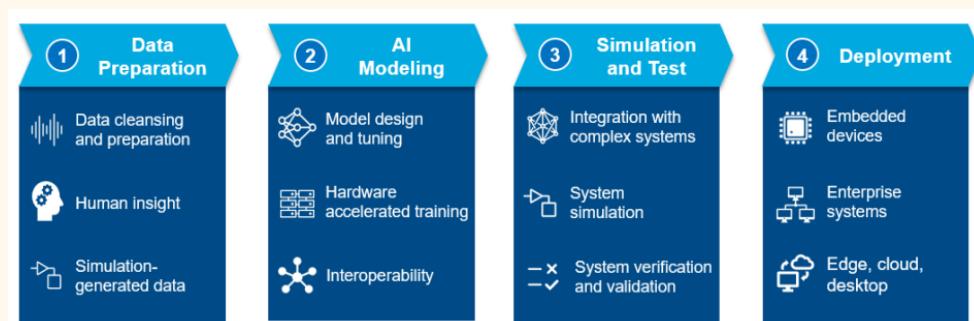
It is a simulation of human intelligence processes performed by machines and computer systems. An artificial intelligence model aims to closely replicate a human brain's decision-making abilities, and tries to **maximize** efficiency and accuracy and **minimize** the loss function (errors).



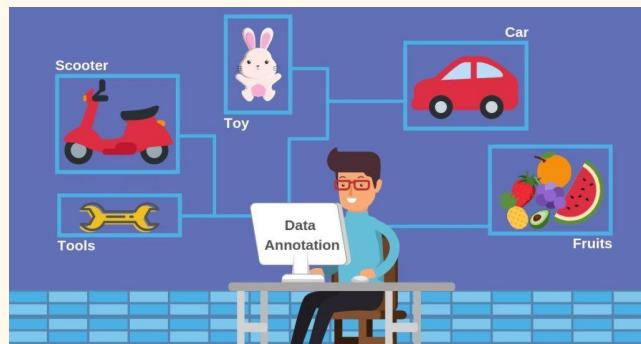
First off, here is a [free course](#) on Coursera that explains the principles of AI in simple terms, taught by the 'Father of AI,' Andrew Ng. This four-week course walks you through important AI terminology, machine learning and deep learning, the workflow of a machine learning project, integrating AI in a company, case studies, AI bias and its role in society, how AI can help develop economies, and so much more.

Now, let's talk about the AI-driven workflow.

For a complete AI-driven workflow we should consider 4 important steps: data preparation, AI modeling, simulation, and testing, and deployment.



Data preparation, arguably one of the most important steps in the AI workflow, includes data cleansing and preparation, human insight, and simulation-generated data. To train an AI model, you should begin with clean, labeled data. It is imperative that you recognize the importance of input data, because wrong, unlabeled data, can lead to significant errors in the AI model. The image below shows clean, labeled data that will be fed into the machine learning model.



You can read more about data preparation [here](#).

In the **AI modeling** stage, the AI model learns from the input data. The goal of this stage is to produce a robust model that makes intelligent, human-like, decisions based on the training data given. Here, you would expect to see the inclusion of machine learning and deep learning, which I will be talking about later. At this stage, it's important to have direct access to many algorithms used for AI workflows, such as classification, prediction, and regression.

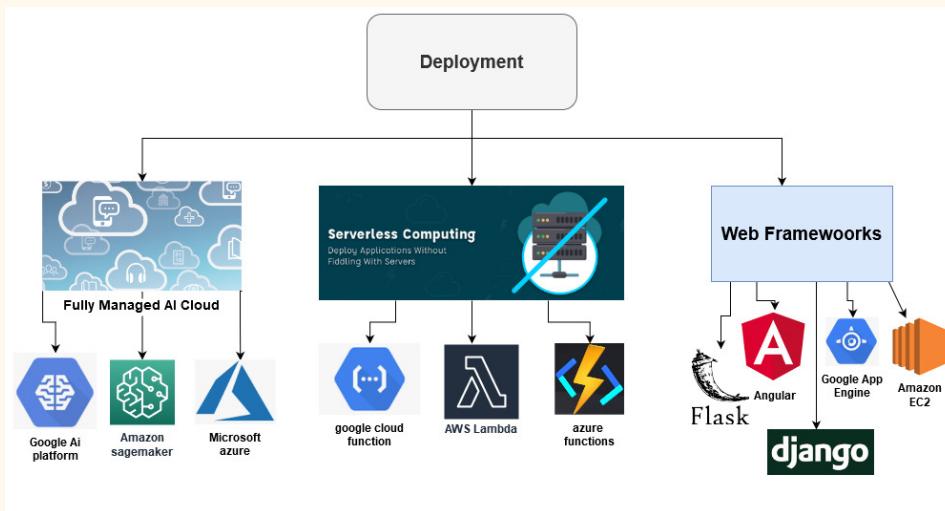
You can read about AI modeling [here](#).

In the next stage, you **test** your model. In this stage, it is very important to repeatedly test your AI model to make sure it is fail-proof. The AI model must have an extremely high accuracy rate in a variety of cases. Simulation and testing for accuracy are crucial to validating that the AI model is working properly, and everything works well together with other systems, before deploying a model into the real world. Let's assume we have an AI model that detects whether a car is present in an image or not. It is important to make sure that the model is tested with numerous different cases: images with a car, images without a car, images with a blurry car, images with half a car being cut off, etc. Only once the model is tested repeatedly will you know its areas of improvement.

You can read more about testing an AI model [here](#).

The final step in the AI workflow is **deployment**. In this stage, you must prepare the model in the final language in which it will be implemented. In this stage, the AI model

must be fit into a designated hardware environment for final use. Below are some ways in which you can deploy a fully functioning AI model.



You can read more about deployment [here](#).

As I mentioned before, machine learning plays a very important role in the field of artificial intelligence. According to an [article](#) by expert.ai, machine learning, “*is an application of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves.*”

Check out this [free course](#) on Coursera by Andrew NG which explains everything you need to know about machine learning.

Now, on to neural networks.

Neural networks are a set of algorithms that have been modeled loosely after the neurons in the human brain. Neural networks, designed to recognize patterns, interpret sensory data through a kind of raw input. The patterns that they recognize are numerical, contained in vectors, into which all real-world data, be it images, sound, text, or time series must be translated. This assortment of algorithms is used in machine learning for data modeling using graphs of neurons.

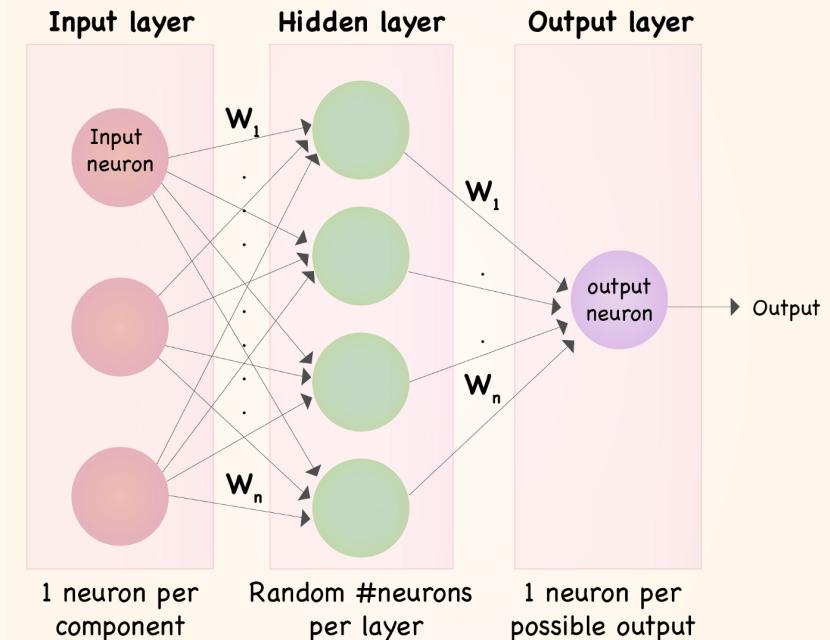
According to an [article](#) by upGrad, here is how neural networks work, in simple terms.

1. Information is fed to the input layer is transferred to the hidden layer
2. The interconnections between the two layers assign weights to each input randomly
3. A bias is added to every input after weights are multiplied with them individually
4. The weighted sum is transferred to the activation function
5. The activation function determines which nodes it should fire for feature extraction
6. The model applies an application function to the output layer to deliver the output
7. The weights are adjusted, and the output is back-propagated to minimize error

The model uses a cost function to reduce the error rate. You will have to change the weights with different training models.

1. The model compares the output with the original result
2. It repeats the process to improve accuracy

The model adjusts the weights in every iteration to enhance the accuracy of the output.



You can read a somewhat more in-depth explanation of how neural networks work [here](#).

To have a more math-based understanding of neural networks, check out this [free online book](#) by Michael Nielsen. The book teaches you about neural networks and deep learning while providing you with numerous real-world examples.

You can also check out this [guide](#) by Karan Jaisingh, which provides high schoolers with the steps and resources to start an in-depth journey into AI and machine learning.

You can also use Google's [Teachable Machine](#) to create a basic AI model without coding anything. Teachable Machine is a web-based tool that makes creating machine learning models fast, easy, and accessible to everyone. If you find difficulty in using Teachable Machine, check out this helpful tutorial [here](#). You can learn more about training the AI model to recognize images [here](#), training the AI model to recognize sounds [here](#), and training the AI model to recognize poses [here](#).

And that brings us to the end of the artificial intelligence unit. Hopefully, with the resources and explanations I have provided, you can begin your high school AI experience and eventually learn to develop application-level AI models.

CYBERSECURITY

Cybersecurity is a term we hear a lot in the field of computer science. Its definition is often boiled down to talking about phishing scams and downloading viruses, but what exactly is cybersecurity?

According to the United States Cybersecurity and Infrastructure Security Agency (CISA), cybersecurity is the “*art of protecting networks, devices, and data from unauthorized access or criminal use and the practice of ensuring confidentiality, integrity, and availability of information.*” The importance of cybersecurity has



skyrocketed in the past decade as the world now relies on computers and the internet in the form of communication (email, smartphones, tablets, etc.), entertainment (video games, social media, apps), transportation (navigation systems), e-commerce (online shopping, credit cards), medicine (medical equipment, medical records), and many more.

Cybersecurity is divided into 2 main categories. The first category includes everything to do with cyberattacks, and the second category includes everything to do with defending against those cyber attacks.

An article by Check Point says that a cyberattack is “*an assault launched by cybercriminals using one or more computers against a single or multiple computers or networks. A cyber attack can maliciously disable computers, steal data, or use a breached computer as a launch point for other attacks.*”

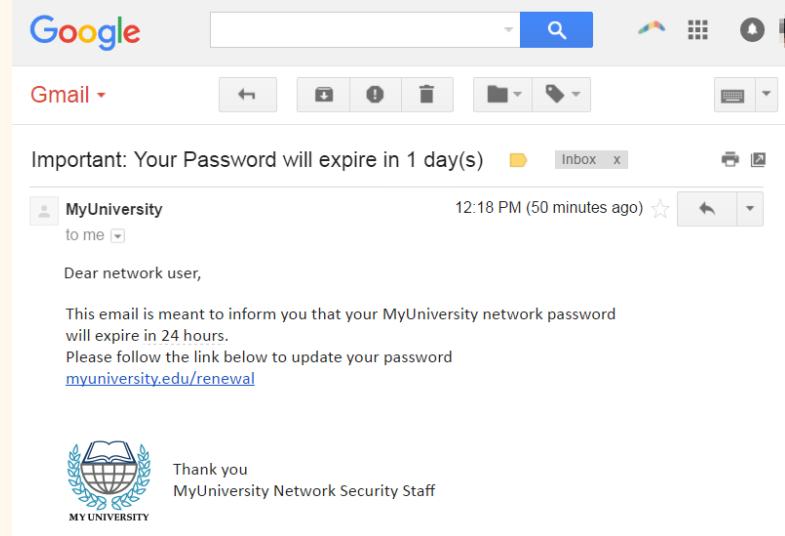
There are many ways in which a cybercriminal can gain unauthorized access to a computer, and it is important to spread awareness of these different cyber attacks. The main cyber attacks are phishing, ransomware, and denial of service.

One of the oldest cyberattacks, dating back to the 1990s, phishing users disguised emails as a weapon. The attackers masquerade as a trusted entity of some kind, often a real or plausibly real person, or a company the victim might do business with. The email recipient is tricked into believing the email is legitimate and is prompted to click a link or download an attachment. Then, a virus is downloaded onto the user's device which wreaks havoc in the user's system, often stealing sensitive data.

You can read more about what phishing attacks are and the characteristics of a phishing attack [here](#).

Here is a real-life example of a phishing attack where Facebook and Google saw the disappearance of \$100 million. According to a Fortune [article](#) written by Jeff John Roberts, in 2013, a 40-something Lithuanian named Evaldas Rmasauskas allegedly hatched a scheme to defraud U.S. tech companies. According to the Justice Department, he forged email addresses, invoices, and corporate stamps to impersonate a large Asian-based manufacturer with whom the tech firms regularly did business. The point was to trick companies into paying for computer supplies. Surprisingly, the scheme worked. Over two years, the corporate imposter convinced accounting departments at the two tech companies to make transfers worth tens of millions of dollars.

Check out [this YouTube video](#) on how to do a phishing attack to gain a better understanding.



Next up, we have a ransomware attack. According to ProofPoint Ransomware is a “*type of malicious software (malware) that threatens to publish or block access to data or a computer system, usually by encrypting it, until the victim pays a ransom fee to the attacker. In many cases, the ransom demand comes with a deadline. If the victim doesn’t pay in time, the data is gone forever.*” In recent years, the frequency of ransomware attacks has skyrocketed with major companies in North America and Europe falling victim to the hands of many cybercriminals.

You can read more about ransomware attacks on the CISA website [here](#).

A real-life example of a ransomware attack is the app CovidLock. This type of ransomware infects a victim’s via malicious files promising to offer more information about the disease.

However, once installed, CovidLock encrypts data from Android devices and denies data access to victims, essentially locking the device. To be granted access to their devices, victims must pay a ransom of USD 100 per device.

Check out this [YouTube video](#) to learn how ransomware attacks add more policy challenges to the US congress’ plates.

And last but not the least, we have denial of service attacks. According to PaloAlto Networks, a denial of service (DoS) attack is “*an attack meant to shut down a machine or network, making it inaccessible to its intended users. DoS attacks accomplish this by flooding the target with traffic or sending it information that triggers a crash. In both instances, the DoS attack deprives legitimate users (i.e. employees, members, or account holders) of the service or resources they expected.*



Signs of a DoS attack



DEGRADATION IN NETWORK PERFORMANCE
Especially when attempting to open files stored on the network or when accessing websites.



SPECIFIC WEBSITE UNAVAILABLE
A particular site does not open or cannot be found.



INABILITY TO ACCESS ANY WEBSITE
All websites are inaccessible on the network.



HIGH VOLUME OF EMAIL SPAM
A higher than usual volume of spam email.

You can read more about DoS attacks [here](#).

Now, a real-life example. According to A10 Networks, Amazon Web Services was hit by a huge DoS attack in February 2020. The attackers targeted an unidentified AWS customer using a technique called Connectionless Lightweight Directory Access Protocol (CLDAP) Reflection. This technique relies on vulnerable CLDAP servers and “amplifies the amount of data sent to the victim’s IP address by 56 to 70 times. The attack lasted for 3 days.

To learn more, check out [this YouTube video](#) on DoS attacks.

As you can see, cyberattacks are quite deadly and can cost users tons of money. However, it is important to recognize that *most* cyberattacks can be prevented.

There are numerous reports of companies that have been attacked by hackers/cybercriminals and lost significant revenue. While these threats look very complex and highly sophisticated to the inexperienced eye, most of them can be prevented by establishing the right security measures.



So, according to an [article](#) by leaf, here are 7 ways that you can protect your systems from cybercriminals.

1. Keep your software and systems fully up to date

Cyber-attacks often occur because many victims’ computer systems are not fully up to date, leaving weaknesses that hackers exploit and gaining access to your network. Once in your device, it is often too late to take preventative action.

To counteract this, it's smart to invest in a patch management system that will manage all your software and system updates, keeping your system resilient and up to date. Other mobile applications like McAfee continuously remind mobile and laptop users to update their software systems.

Check out McAfee [here](#). McAfee also offers numerous other antivirus features, so you can explore their website to see how these work to gain a deeper understanding of cybersecurity.

2. Ensure Endpoint Protection

According to McAfee, “*endpoint security is the practice of securing endpoints or entry points of end-user devices such as desktops, laptops, and mobile devices from being exploited by malicious actors and campaigns. Today’s endpoint protection systems are designed to quickly detect, analyze, block, and contain attacks in progress.*”

Check out ESET endpoint protection [here](#). Their website further explains how endpoint protection works and how it helps protect users and companies.

3. Install a Firewall

Based on a defined set of security rules, a firewall, which is a security network that monitors incoming and outgoing network traffic, decides whether to allow or block specific traffic. This way, a Firewall can protect your device from blocking any malware from entering your system.

Again, [McAfee](#) offers excellent Firewall protection. You can read more about how Firewalls work on their website.

4. Backup your data

In the event of a cyberattack, it is important to have your data backed up to avoid serious downtime, loss of data, and serious financial loss.



Different ways to backup your data are using a USB stick, an external hard drive, etc.

5. Control access to your systems

One entry into your computer systems could be physical. In a public workplace, someone could stick a USB stick into your computer port without your knowledge. This could download some malware or unwanted software.

6. Wifi security

Connecting your computer/electronic device to a secured Wifi network is very important because it prevents hackers from accessing your wireless network and stealing data using your Wifi.

7. Passwords

This one is quite obvious, but it's important to change around passwords with different accounts to prevent a hacker from accessing all accounts at once.

Now, this is just the tip of the iceberg. There are so many topics under cybersecurity that I haven't covered in this section of the guide, however, I will be providing you with resources to help you discover (if you have one) your path in cybersecurity.

Check out this [free cybersecurity course](#) for beginners offered by IBM on Coursera.

You can also check out this [free cybersecurity fundamentals course](#) offered by the Rochester Institute of Technology on edX.

Onto the more specific courses.

[Here](#) is a free online course offered by UC Irvine on Coursera. In the course, students are introduced to the field of cybersecurity with a focus on the domain of security and risk management. Topics include the fundamentals concepts and goals of cybersecurity (the CIA triad), security governance design, the NIST cybersecurity framework, relevant laws and regulations, and the roles of policies, strategies, and procedures in cybersecurity governance.

You can find another free course offered by Saint Petersburg State University on Coursera [here](#). This is a great course that gives you an all-around understanding of cybersecurity, social media, big data, and how all are used and manipulated by governments and other groups to influence population groups.

Next, we have a [free online course](#) offered by the Rochester Institute of Technology on edX. The course is about digital forensics, which involves the investigation of computer-related crimes to obtain evidence to be presented in a court of law. In the course, you will learn the principles and techniques for digital forensics investigation and about core forensics procedures to ensure court admissibility of evidence. You will learn how to perform a forensic investigation and will be guided through forensic procedures and review and analyze forensic reports.

You can find a free online course offered by MIT on edX [here](#). This course will prepare anyone who wants to work with agencies that are worried about their vulnerability to cyberattacks. Through a series of lectures, case studies, simulations, debriefings, and short assigned readings, you will learn what cities can and should do to reduce their vulnerabilities. The course also includes checklists of various kidneys that cybersecurity vulnerability assessors need to ask and answer.

The next free online course, offered by the Rochester Institute of Technology on edX, can be found [here](#). In this course, you will examine the various areas of network security including intrusion detection, evidence collection, and defense against cyber attacks. The issues and facilities available to both the intruder and data network administrator will also be examined to illustrate their effect. You will learn the principles and concepts of wired and wireless data network security and will be guided through a series of labs and experiments to explore various mechanisms for securing data networks including physical layer mechanisms, filters, applications, and encryption. You will also analyze attack/defend scenarios and determine the effectiveness of particular defense deployments against attacks.

[Here](#) is a free online course offered by the Rochester Institute of Technology on edX. This course will learn the key principles of risk analysis, risk assessment, and risk mitigation for information security using qualitative and quantitative methodologies.

And finally, check out this [short, 40 minute YouTube tutorial](#) that covers coding for cybersecurity. It covers cybersecurity projects and stresses the importance of python in cybersecurity. You can skip the first half of the tutorial if you wish because it covers the fundamentals of cybersecurity which you have already covered.

Finally, we have reached the end of our cybersecurity unit. Once again, I hope the resources I have provided you with are helpful enough for you to begin your high school cybersecurity experience, and hopefully help you discover whether cybersecurity is a topic worth pursuing in college.

WEB DEVELOPMENT/DESIGN

Have you ever wanted to build a website? Maybe for fun, or maybe for a potential business idea that you were thinking of? Maybe to promote something or maybe to write blogs? Well, this section of the guide will help you begin your journey in web development, a very important part of computer science that hasn't gained enough recognition in high school computer science courses.



According to Techopedia, web development refers in general to “*the tasks associated with developing websites for hosting via intranet or internet. The web development process includes web design, web content development, client-side/server-side scripting, and network security configuration, among other tasks. Web development encompasses all the actions, updates, and operations required to build, maintain and manage a website to ensure its performance, user experience, and speed are optimal.*” Web development might also “include all those strategic actions needed to ensure its proper ranking on search engine results. Usually, those tasks pertain to a different specialization, namely search engine optimization.”

In the world of web development, there are three types of people.

The first is the front-end developer, also known as the “web developer/designer”. According to frontenddevelopers.com Front-End web development, also known as client-side development is *“the practice of producing HTML, CSS, and JavaScript for a website or Web Application so that a user can see and interact with them directly. The challenge associated with front-end development is that the tools and techniques used to create the front end of a website change constantly and so the developer needs to constantly be aware of how the field is developing.”* The objective of designing a site is to ensure an aesthetic, but efficient user experience. The difficulty of this task increases with varying screen sizes and resolutions. As a result, designers need to make sure that their website is perfect in different browsers, different operating systems, and different devices, which requires careful planning on the developer’s side. As a front-end web developer, it is crucial to be fluent in JavaScript, CSS, and HTML. JavaScript is a text-based programming language used both on the client-side and server-side that allows you to make web pages interactive. HTML and CSS, on the other hand, are languages that give structure and style to web pages.

Here are some courses that can help you begin front-end web development.

You can find an introduction to JavaScript course on YouTube [here](#). The course teaches you everything you need to know about JavaScript.

You can find an introduction to HTML course on YouTube [here](#). The course teaches you everything you need to know about HTML.

You can find an introduction to CSS course on Youtube [here](#). The course teaches you everything you need to know about CSS.

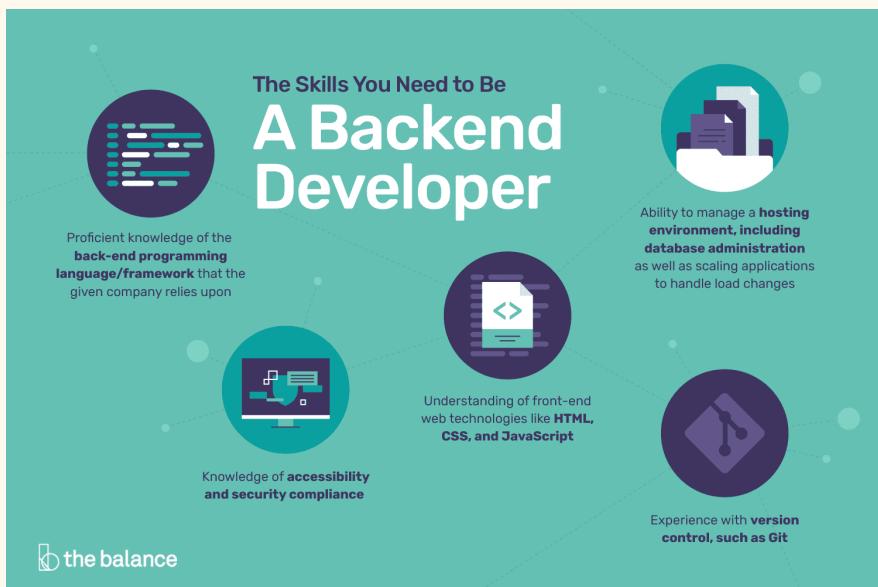
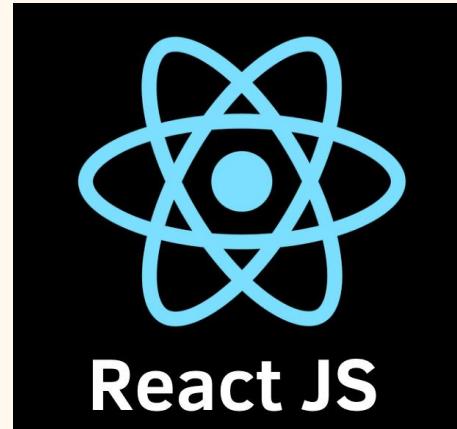
You can find a free online course offered by the Coursera Project Network on Coursera [here](#). In this short course, you will learn about React Basics & Theory, Initialization, React Router, Components, Hooks, State, and deployment with Netlify. You should know basic HTML & CSS before beginning this project.

In case you didn't know, ReactJS is an open-source JavaScript library that is used for building user interfaces specifically for single-page applications. It's used for handling the view layer for web and mobile apps. React also allows us to create reusable UI components.

You can also find a free online course offered by IBM on Coursera [here](#). In this course, you will understand the Cloud Development Ecosystem and Terminology like front-end developer, back-end, server-side, full-stack, and more. You will also become familiar with the developer tools and IDEs used by web developers and work with programming languages used by front-end developers for creating user interfaces. You will also learn how to practice and develop hands-on skills to work with HTML, CSS, and JavaScript.

The second type is the back-end developer, also known as the person no one really knows about. According to Toptal, a back-end web developer is *"responsible for server-side web application logic and integration of the work front-end developers do."*

Back-end developers usually write the web services and APIs used by front-end developers." The most popular back-end programming languages include PHP, Ruby, Python, Java, Rust, C#, and more.



In the image below, you can read through a company's requirements for a back-end developer.

Responsibilities

- Integration of user-facing elements developed by a front-end developers with server side logic
- Building reusable code and libraries for future use
- Optimization of the application for maximum speed and scalability
- Implementation of security and data protection
- Design and implementation of data storage solutions

Skills And Qualifications

- Basic understanding of front-end technologies and platforms, such as JavaScript, HTML5, and CSS3
- Good understanding of server-side CSS preprocessors, such as LESS and SASS
- Understanding accessibility and security compliance **[[Depending on a specific project]]**
- User authentication and authorization between multiple systems, servers, and environments
- Integration of multiple data sources and databases into one system
- Management of hosting environment, including database administration and scaling an application to support load changes
- Data migration, transformation, and scripting
- Setup and administration of backups
- Outputting data in different formats
- Understanding differences between multiple delivery platforms such as mobile vs desktop, and optimizing output to match the specific platform
- Creating database schemas that represent and support business processes
- Implementing automated testing platforms and unit tests
- Proficient knowledge of a back-end programming language **[[Depending on the specific case, a developer should have the knowledge of one or more of PHP, Python, Ruby, Java, .NET, JavaScript etc.]]**
- Proficient understanding of code versioning tools, such as Git
- Proficient understanding of OWASP security principles
- Understanding of "session management" in a distributed server environment

To get a better understanding of what backend developers do, watch [this short YouTube video](#).

You can find an introduction to PHP course on YouTube [here](#). The course teaches you everything you need to know about PHP.

You can find an introduction to Ruby course on YouTube [here](#). The course teaches you everything you need to know about Ruby

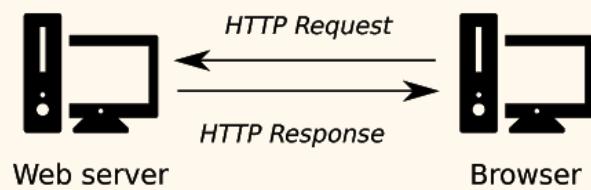
You can find an introduction to Rust course on YouTube [here](#). The course teaches you everything you need to know about Rust.

You can find an introduction to C# course on YouTube [here](#). The course teaches you everything you need to know about C#.

For further clarification, you can watch a short YouTube video on the differences between back-end and front-end development [here](#).

Finally, we have the third type... the full-stack developer, also known as the superhero. According to MADCAP, a full-stack developer is “*a unique developer type as they can program and work on both front-end and back-end parts of a website. They can literally create an entire function dynamic website themselves (if they choose) as they can understand both ends of web development. Full-stack developers tend to have prowess in coding in multiple programming languages and their knowledge and experience tend to be a mile-wide. Because of this, they generally work well with both the front-end and back-end developers during the development process.*” A full-stack developer is a person who can develop both client and server software. In addition to mastering HTML and CSS, they should also know how to:

- Program a browser (using JavaScript, jQuery, Angular, or Vue)
- Program a server (using PHP, ASP, Python, or Node)
- Program a database (using SQL, SQLite, or MongoDB)



Now, you might ask, what is a database, and how do you program one?

A database is a place where you can store information related to a specific topic in an organized manner. According to Oracle, a database is “*an organized collection of structured information, or data, typically stored electronically in a computer system. The data can then be easily accessed managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.*” A database query is a similar action that is most clearly associated with some sort of CRUD (create, read, update, delete) function. A database query is a request to access data from a database to manipulate it or retrieve it. This allows us to perform logic with the information we get in response to the query.



There are many different types of databases. These include centralized databases, cloud databases, commercial databases, distributed databases, end-user databases, graph databases, NoSQL databases, object-oriented databases, etc.

Databases are a very important part of modern life. Without them, most computer functions would cease to exist. It is important to understand the different types of databases that exist and how they should be used if you are someone who relies on storing information within a computer.

[Here](#) is an introduction to SQL course offered by UC Davis on Coursera.

[Here](#) is an intermediate course that introduces you to the basics of Django, Bootstrap, and SQL offered by IBM on edX.

[Here](#) is a full database course for beginners on YouTube.

[Here](#) is an introduction to user experience course offered by HEC Montreal on Coursera.

[Here](#) is an introduction to web accessibility course offered by W3C on Coursera.

As a high schooler, I think it's important to know web development on both the client-side and the server-side. I would recommend searching for complex website templates, then downloading the codes, and understanding the JavaScript, CSS, and HTML behind each website



That brings me to the end of the web development unit. Once again, I hope the resources I have provided you with are helpful enough for you to begin your high school web development experience, and hopefully code several complex websites.

SOFTWARE/APP DEVELOPMENT

Here begins our journey into the world of software and application development. An app is a way to bring your consumer closer to your products or services. It allows customers to easily interact and make transactions without physical limitations. To have positive results in an app, the app must have a focus on user usability and navigability.



An application, also referred to as an application program or application software, is a software package that performs a specific function directly for an end-user or, in some cases, for another application. An application can be self-contained or a group of programs.

There are 6 different stages to the software development cycle

1. Planning

All software development projects begin here, the place where the initial idea for the software is formed. In this phase, you must focus on identifying the problem, gathering information needed to plan a solution and a review of all the available data. In this phase, it is important to gather enough information to decide how to proceed. This is also known as the brainstorming phase.

2. Analysis

This stage is essentially a feasibility study to see if your idea is viable. The goal is to reevaluate the proposed solution from a consumer perspective. You will have to build out the rest of the idea and find ways to justify its development. Much of the work in this step will outline the design and development stages later.

3. Design

In this stage, you create the fully developed design of your software. Here, all the design work happens so that the project can enter its development phase. In many cases, systems for further development and coordinating resources can be set up at this stage. At the end of this design process, you should have a useable plan for the software including how it looks, how it functions, and how it will be built.

4. Implementation

Here, the software is assembled. This involves a variety of processes, including coding, setting up infrastructure, and creating documentation on how the system works. Developers may work with designers to ensure that their work aligns with the designs.

5. Testing and Integration

When the bulk of the work is completed, it is important to test your product. All software must be rigorously tested before being released to the public so that you can spot any errors and fix them immediately.

6. Maintenance

After the application has finally been tested and released to the public, it is important to continuously update the software and add new features and improvements.

Now that you know the software development cycle, you can start coding your first app.

But first, let's go over Spinx's top 15 programming languages for app development.

1. JavaScript
2. Kotlin

You can find a free tutorial on YouTube [here](#).

3. C++

You can find a free tutorial on YouTube [here](#).

4. C#
5. Python
6. PHP
7. Swift

You can find a free tutorial on YouTube [here](#).

8. Objective-C

You can find a free tutorial on YouTube [here](#).

9. Java
10. HTML 5
11. Ruby
12. Rust
13. Lua

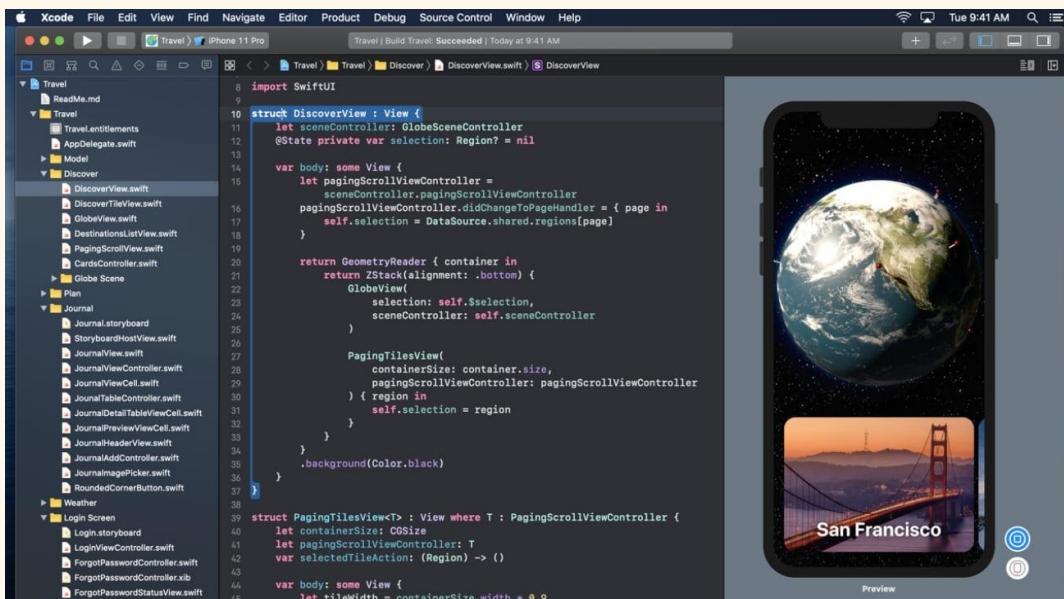
You can find a free tutorial on YouTube [here](#).

14. Action Script

You can find a free tutorial on YouTube [here](#).

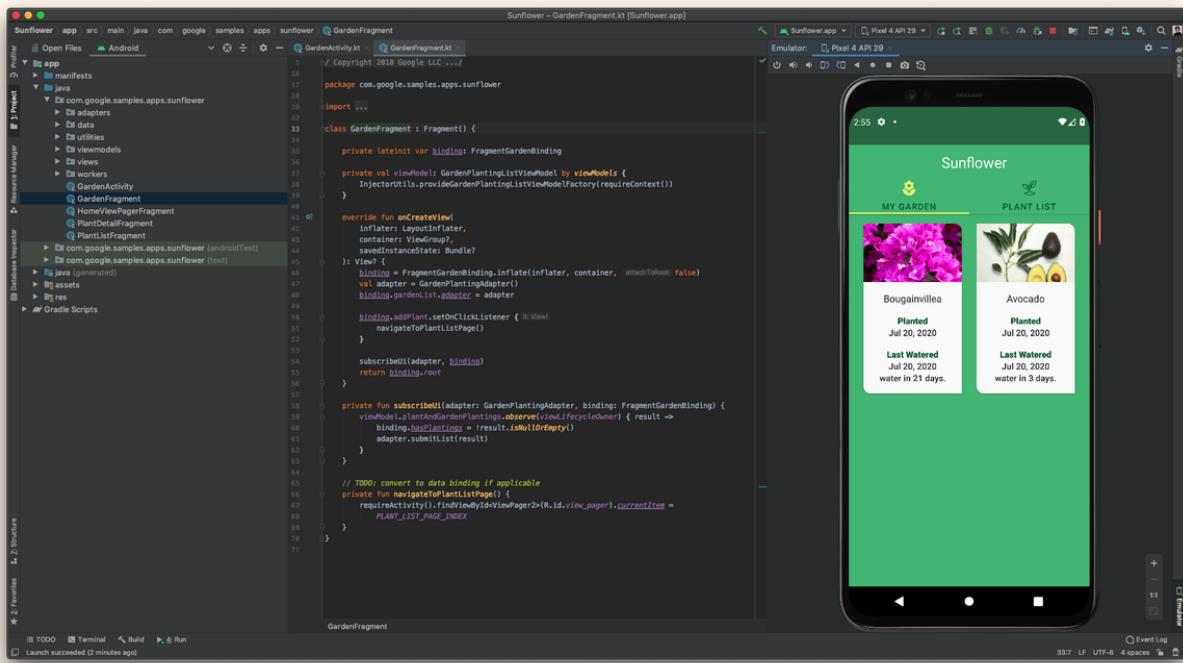
15. SQL

In this section of the guide, however, I will be focusing on android and iOS app development. In iOS application development, we make mobile apps for Apple hardware, including iPhones, iPads, and the iPod Touch. The software is written in Swift or Objective-C and then deployed to the App Store for users to download. The software I use for iOS application development is XCode. XCode is Apple's integrated development environment for macOS, used to develop software for macOS, iOS, iPadOS, watchOS, and tvOS. It was first released in 2003; the latest stable release is version 12.5.1, released on June 21, 2021, and is available via the Mac App Store free of charge for macOS Big Sur users.



You can learn more about XCode and download it [here](#).

For Android development, we have Android Studio, Android's official IDE. It is purpose-built for Android to accelerate your development and help you build the highest-quality apps for every Android device. Developed by Google and JetBrains, Android Studio was released in December 2014 and is in its latest version (2020.3.1), released in May.



You can learn more about Android Studio and download it [here](#).

These apps provide you with a variety of features that can help you build your first application. However, before beginning, I think it's important to start with the basics and go over the resources I have provided.

In addition, you can check out these courses.

You can find an introduction to Android Studio course on YouTube [here](#). In this course, you can learn how to develop an android app from scratch in this full course for beginners. In this course, you will learn how to set up an environment, create your first application, learn about variables and arithmetic operators, learn about relational and operators, learn about loops and arrays, learn about OOPS, learn about interfaces and abstract classes, learn how to design in Android Studio, and learn about much much more.



You can find an introduction to XCode course on YouTube [here](#). In this iOS tutorial, you'll learn how to read and write Swift code so that you can build your very own app using XCode. By the end of this iOS development tutorial, you'll have built your first fully functional app.

For more formal tutorials check out the following links below.



[Here](#), you can find a free Android app development course on Coursera. This specialization enables learners to successfully apply core Java programming languages features and software patterns needed to develop maintainable mobile apps comprising core Android components, as well as fundamental Java I/O and persistence mechanisms. The Capstone project will integrate the material from throughout the Specialization to exercise and assess the ability of learners to create an interesting Android app by applying knowledge and skills learned in previous MOOCs, including Java programming features, Android Studio tools, Android Activity components, Material Design, file I/O, and data persistence, unit testing, and software patterns. The project itself will be similar in design goals to previous assignments, however, it will provide less of the skeleton code than earlier MOOCs provide to enable more creativity to learners and greater opportunity for learners to customize the app.

[Here](#), you can find a free XCode course on Coursera. This Specialization covers the fundamentals of iOS application development in the Swift programming language. You'll learn to use development tools such as XCode, design interfaces and interactions, evaluate their usability, and integrate camera, photo, and location information to enhance your app. In the final Capstone Project, you'll apply your skills to create a fully functioning photo editing app for iPhone, iPad, and Apple Watch. A Mac computer is required for success in this course.

This brings us to the end of our software development section. In the next unit, I will cover potential projects for you to begin to get started on your computer science application journey.

REAL-WORLD APPLICATION

Now, with all that you have learned throughout this guide, you should be able to develop several complex software that have some use in the real world. In this final section of the guide, I will provide you with a detailed list of applications and software you can build to help make an impact on the world, and strengthen your college profile. For each topic covered in this guide, I will provide you with 2-3 projects.



Let's dive right in.

The first topic is artificial intelligence. Ranging from finance to sports, security to healthcare, agriculture to infrastructure, and more, there is an endless number of projects that you can work on. Here are a few examples.

1. Computer vision
 - a) Distracted drivers

In a project like this, you would use computer vision to improve road safety by detecting distracted drivers. You would use a variety of ML techniques, including convolutional neural networks and transfer learning to create and improve effective models. Remember to improve your model by analyzing when it succeeds and when it fails, by creating saliency maps

to make your model impenetrable, or by editing your dataset to preserve privacy.

b) Facial emotion detection

In a project like this, you would use computer vision models to detect emotions on people's faces, helping provide feedback in places like online education. First, you would create hand-crafted strategies for identifying emotions based on facial landmarks. Then you would create and analyze machine learning models using convolutional neural networks to learn emotions automatically. You would use techniques like dimensionality reduction and transfer learning to make your models as accurate and useful as possible.

c) Object detection from street images

In this project, you would create computer vision models to help self-driving cars 'see' by identifying objects in street images. Using a variety of image manipulation techniques and convolutional neural network architectures, you would locate and identify objects like traffic lights, cars, and people. Finally, you would use a YOLO model to perform object detection on videos in real-time, creating systems that are fast enough to use in autonomous vehicles!

2. Recommender systems

a) Music recommender system

In this project, you would work with Apple Music and Spotify music data. You would review logistic regression to predict which songs will become top hits based on different song metrics. Then you would work on a system to recommend new songs to listeners (using popular songs and similar songs to ones you've listened to). Finally, you would return to predicting top hits, but this time by using the raw music audio.

b) Movie recommender system

In this project, you would work with Netflix and Amazon Prime movie data. You would review logistic regression to predict which movies will increase in popularity based on different movie metrics. Then you would work on a system to recommend new movies to viewers (using popular movies and similar movies to ones you've watched).

3. AI + ethics

a) Resume bias

In this project, you would learn how to implement 'fair' AI using automated recruiting as an example. Along the way, you would train/tune an accurate model, and adapt this model to be fair with a variety of techniques.

b) Criminal justice

In this project, you would discover hidden racial biases in machine learning systems used today for deciding who to send to jail. As you create and analyze fairer models, you will build your skills in machine learning and probability. You will also learn more about the ethical questions of AI.

4. Healthcare

a) Pneumonia detection

In this project, you would create a computer vision system to help diagnose pneumonia from chest X-rays. You'll review neural networks and create convolutional neural networks in Keras to process medical images. You could then improve your models using transfer learning: adapting from expert models. This model could even be deployed to an app.

b) Skin cancer detection

In this project, you would use computer vision and image classification tools to develop machine learning models to diagnose skin lesion images. By using convolutional neural networks (CNN), you could create a model

that saves lives. It is important to study your model for bias and make sure that it works for people with a variety of skin tones.

5. Natural Language Processing (NLP)

a) Anti-refugee sentiment analysis

In this project, you would help prevent hate speech on the internet by identifying anti-refugee tweets. First, you would create a rule-based classifier to identify hateful tweets. You would then train a model on a dataset of real tweets, improving it with a variety of NLP techniques including modern methods like word embeddings. Your model could end up making the internet a safer place.

b) Disaster-relief improvement

In this project, you would help relieve natural disasters by identifying tweets from people asking for help or offering resources like food, water, or shelter. You would first create rules-based classifiers to classify different posts. You would then train a model on a dataset of real tweets, improving it with a variety of NLP techniques including modern methods like word embeddings. Your model could help first responders connect people in need with people who can help.

c) Fake news detection

In this project, you would create machine learning models to identify fake news websites, helping internet users find reliable sources of information. You would apply NLP techniques including word embeddings to analyze the websites' content. You can also create a model to generate fake news.!

d) Financial tweet analysis

In this project, you would use logistic regression, LSTMs, and BERT to analyze the sentiment of financial tweets. The ultimate goal is to predict long-term stock market trends, which can help us manage our wealth and anticipate potential booms and busts.

6. Astronomy

a) Searching for exoplanets with AI

In this project, you would use data collected from NASA's Kepler space telescope to train AI models to detect exoplanets. Finding exoplanets could help us discover alien life.

7. AI + genomics

a) Trace and track Covid-19 with Alga

In this project, you would trace the origins of COVID-19 to help understand its spread. By working with genomic data, you would create a machine learning model to identify the geographic origins of different COVID-19 strains. Finally, you would apply advanced techniques like dimensionality reduction for building more accurate models from complex biological datasets.

In addition to these projects, I believe that chatbots are one of the most versatile AI projects because they can be implemented in almost every field. Today, chatbots are most commonly used in the customer service space for companies, but I believe they can have applications in many other ways.

1. Customer service chatbot

[Here](#) is a video on YouTube that explains everything about chatbots and their relation to AI.

2. 'Friend' chatbot

[Here](#) is a short YouTube tutorial on how to build an intelligent AI chatbot using Python.

3. Suicide prevention chatbot

[Here](#) is an article that you can read about suicide prevention chatbots.

With facial recognition, there are tons of projects that can be created.

1. Record attendance at virtual meetings
2. Find missing people
3. Recognize VIPs at events
4. Verify identity at ATMs
5. Prevent cheating in a school environment

Now let's move on to cybersecurity projects.

Here are 3 cybersecurity projects you can start today.

1. Keylogging

A Keylogger is software that can identify the keystrokes made in a particular system. A keystroke is just any interaction you make with a button on your keyboard. Keystrokes are how you 'speak' to your computers. Each keystroke transmits a signal that tells your computer programs what you want them to do.

Check out [this YouTube video](#) that teaches you how to build a Keylogger with Python.

2. Packet Sniffing

Packet sniffing is a technique whereby packet data flowing across the network is detected and observed. Network administrators use packet sniffing tools to monitor and validate network traffic, while hackers may use similar tools for nefarious purposes.

Check out [this tutorial](#) on how to build a Packet Sniffer

3. Bug Bounties

Bug Bounties is a software tool that helps you find bugs in code.

Check out [this YouTube video](#) that teaches you how to build a Bug Bounty

Check out [this video](#) that tells you about cybersecurity projects and creating plans for them.

You can also check out [this video](#) that covers 5 cybersecurity projects for beginners.

Under web and app development, there are thousands of things you can build based on day-to-day problems that you encounter. Websites include blogs, web apps, and so much more. Apps include games, information apps, guides, entertainment apps, etc. A web app, website, or app depends on a problem that you can encounter in your day-to-day lives, but I would recommend starting with games because it has a healthy balance of graphics and user interface systems.

Check out [this YouTube tutorial](#) on building Flappy Bird using JavaScript.

[Code For Social Good](#) is also a place to apply the things you have learned throughout this guide.

CONCLUSION

Through the course of this guide, I have provided you with the tools you need to begin your exponential growth in the field of computer science. Computer science is a vast subject, much of which is not covered in your average CS class, and I believe we all must have a deeper level of understanding in the fields of the future. Though I have not covered all the CS topics (IoT etc.), hopefully, the information and the resources I have provided will aid you in your high school computer science journey.