



GREAT NECK PUBLIC SCHOOLS

"Where Discovery Leads to Greatness"

#DiscoverGreatness

Phipps Administration Building
Office of Instruction
345 Lakeville Road
Great Neck, New York 11020

516.441.4009
516.441.4010
Fax 516.441.4992

Curriculum Profile: AP Computer Science Principles

<u>Department</u>	Mathematics	
<u>Course Name</u>	AP Computer Science Principles	
<u>Course Length</u>	Full Year	
<u>High School Credits</u>	1	
<u>Description</u>	The main goal of this course is to expose students to the massive field of Computer Science with a focus on non-programming areas. While the course does involve some coding elements – pseudocode (not script language not tied to any real programming language) will be the primary tool used. The “Big Ideas” of the course include creative development, data, algorithms and programming, computer systems and networks, and the impact of computing. All students take the AP exam in May – which consists of a traditional multiple-choice exam, as well as a project portfolio that is due prior to the written exam.	
<u>Target/eligible students</u>	Grades 10-12, must be enrolled in or have completed Geometry.	
<u>Learning Standards Link(s)</u>	AP Computer Science Principles Course and Exam Description, Effective Fall 2020 (collegeboard.org) Pages 29-125	
<u>Primary texts and materials</u>	Provided by teacher	
<u>Scope/Sequence</u>	Overview /Standards	Students will: learn appropriate research techniques for improving interfaces/software while respecting and protecting test subjects and data, how the internet works from a conceptual and technical standpoint, cybersecurity attacks and their aftereffects, graph traversal algorithms to facilitate network communication and data transfer, big data analysis and how to clean/treat data fairly and appropriately, digital divide caused by access or no access to technology/internet access, introductory programming in both pseudocode and a web based platform, control structures, AP exam project component (40% of AP Score), AP Exam review, and a class specific final post-AP project.
<u>Unit 1: Human-Computer Interaction</u>	- DAT-2.C.4 - DAT-2.C.5 - CRD-1.A.4	This unit investigates the intersection between cognitive science, psychology, and software/interfaces. Time is spent learning about Don Norman who is a pioneer in all 3 of these fields and his contributions to human centered design. Next, we investigate the Design Life Cycle – a 4 phase conceptual model for how to appropriately investigate interfaces and their users to determine potential changes that could improve the interfaces' ability to help users complete their tasks. A special focus is put on proper research methodology, data collection, analysis, and reflecting on the entirety of the students work during this time. - Cleaning data is a process that makes the data uniform without changing their meaning (e.g., replacing all equivalent abbreviations, spellings, and capitalizations with the same word). - Problems of bias are often created by the type or source of data being collected. Bias is not eliminated by simply collecting more data - Collaboration that includes diverse perspectives helps avoid bias in the development of computing innovations.

	• CRD-2.F.1 • CRD-2.F.3 • CRD-2.F.7	• The design of a program incorporates investigation to determine its requirements. • Some ways investigation can be performed are as follows: ○ collecting data through surveys ○ user testing ○ interviews ○ direct observations • The design phase of a program may include: ○ brainstorming ○ planning and storyboarding ○ organizing the program into modules and functional components ○ creation of diagrams that represent the layouts of the user interface ○ development of a testing strategy for the program
<u>Unit 2: Internet & Cybersecurity</u>	• CSN-1.B.1 • CSN-1.B.3 • CSN-1.B.6 • CSN-1.C.2 • IOC-2.A.1 • IOC-2.A.11 • IOC-2.B.8	This unit introduces students to the idea of a network – which allows two or more nodes (devices) to communicate with each other. Over time as these networks grew and connected with one another – we formed what we now call the Internet. Because there are countless different types of devices and information, a system must exist that facilitates the inclusion of everything and anything. This leads into the study of protocols, the OSI model, shortest path algorithms, etc. At the end of the unit students should now have a strong technical understanding of networking as well as the pros and cons of the current systems. Once the foundation is in place – we can study cybersecurity and how many of the issues we deal with today (identity theft, DOS attacks) stem from issues in internet protocols, human error, and more. • The Internet is a computer network consisting of interconnected networks that use standardized, open (nonproprietary) communication protocols. • A protocol is an agreed-upon set of rules that specify the behavior of a system. • The scalability of a system is the capacity for the system to change in size and scale to meet new demands. • Packets contain a chunk of data and metadata used for routing the packet between the origin and the destination on the Internet, as well as for data reassembly • Personally identifiable information (PII) is information about an individual that identifies, links, relates, or describes them. Examples of PII include: ○ Social Security number ○ age ○ race ○ phone number(s) ○ medical information ○ financial information ○ biometric data • Information placed online can be used in ways that were not intended and that may have a harmful impact. For example, an email message may be forwarded, tweets can be retweeted, and social media posts can be viewed by potential employers. • A computer virus is a malicious program that can copy itself and gain access to a computer in an unauthorized way. Computer viruses often attach themselves to legitimate programs and start running independently on a computer.

<u>Unit 3: Data/Algorithms</u> (<u>Graph Theory, Compression, Encryption</u>)	• DAT-1.D.1 • DAT-1.D.3 • DAT-2.A.4 • IOC-2.B.5 • CSN-2.A.7 • AAP-4.A.2	Data is not the easiest idea to understand – as it is a digital representation of information. The goal of this unit is to understand what form digital data takes, how it can be compressed or expanded to facilitate the travel and storage of said data, how to read and interpret data in many forms, how data moves from Point A to Point B efficiently, and how encryption algorithms work to protect data. • Data compression can reduce the size (number of bits) of transmitted or stored data. • The amount of size reduction from compression depends on both the amount of redundancy in the original data representation and the compression algorithm applied. • Often, a single source does not contain the data needed to draw a conclusion. It may be necessary to combine data from a variety of sources to formulate a conclusion. • Encryption is the process of encoding data to prevent unauthorized access. Decryption is the process of decoding the data. Two common encryption approaches are: ○ Symmetric key encryption involves one key for both encryption and decryption. ○ Public key encryption pairs a public key for encryption and a private key for decryption. The sender does not need the receiver's private key to encrypt a message, but the receiver's private key is required to decrypt the message. • The “speedup” of a parallel solution is measured in the time it took to complete the task sequentially divided by the time it took to complete the task when done in parallel. • A decision problem is a problem with a yes/no answer (e.g., is there a path from A to B?). An optimization problem is a problem with the goal of finding the “best” solution among many (e.g., what is the shortest path from A to B?).
<u>Unit 4: Stock Market Analysis (Big Data)</u>	• DAT-1.A.2 • DAT-1.A.5 • DAT-2.A.2 • DAT-2.A.4	Now that students are familiar with the basics of data – we explore it head on. The stock market is a complex system of data – where analysts and artificial intelligence parse through nearly infinite data to try and make predictions and succeed in the stock market. Students will explore how to organize massive amounts of data, how to filter it, how to draw conclusions and predict trends mathematically and appropriately. • Computing devices represent data digitally, meaning that the lowest-level components of any value are bits. • Abstraction is the process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea. • Data provide opportunities for identifying trends, making connections, and addressing problems. • Often, a single source does not contain the data needed to draw a conclusion. It may be necessary to combine data from a variety of sources to formulate a conclusion

<u>Unit 5: Digital Divide/Computing Ethics</u>	• IOC-1.C.2 • IOC-1.C.5 • IOC-1.F.1 • IOC-1.F.5	Many students take Internet and technology access for granted. Not only do certain regions of the world have limited or no access – but the gap between those who do and do not continue to grow as technology advances. Students will conduct research to investigate why this happens – and what we can do in the future to try and close this gap. • The “digital divide” refers to differing access to computing devices and the Internet, based on socioeconomic, geographic, or demographic characteristics. • The digital divide is affected by the actions of individuals, organizations, and governments. • Material created on a computer is the intellectual property of the creator or an organization. • Some examples of legal ways to use materials created by someone else include: ○ Creative Commons—a public copyright license that enables the free distribution of an otherwise copyrighted work. This is used when the content creator wants to give others the right to share, use, and build upon the work they have created. ○ open source—programs that are made freely available and may be redistributed and modified ○ open access—online research output free of any and all restrictions on access and free of many restrictions on use, such as copyright or license restrictions
<u>Unit 6: Programming: Intro to App Design</u>	• AAP-1.A.1 • AAP-1.D.1 • AAP-2.B.1 • AAP-2.C.1 • AAP-2.G.1	Algorithms are a powerful tool – but at some point, we need to take them off paper and take advantage of technology to turn them into a tangible element. The first half of the programming unit in this course focuses on how basic algorithms can be turned into “code”. The AP course has a specific pseudocode language that is used to allow students to focus on the “why” and not on the syntax and minor details that exist in most traditional languages. Students explore the ideas of variables and data storage, ordering of steps in algorithms/code, basic arithmetic operations and simple logic. • A variable is an abstraction inside a program that can hold a value. Each variable has associated data storage that represents one value at a time, but that value can be a list or other collection that in turn contains multiple values. • Data abstraction provides a separation between the abstract properties of a data type and the concrete details of its representation. • Sequencing is the application of each step of an algorithm in the order in which the code statements are given. • Arithmetic operators are part of most programming languages and include addition, subtraction, multiplication, division, and modulus operators. • Selection determines which parts of an algorithm are executed based on a condition being true or false.
<u>Unit 7: Programming: Advanced Analysis (Pseudocode)</u>		The advanced section builds upon Unit 6 – but now we investigate more conceptual structures, such as iteration, functions, APIs, and data structures. Again – the focus here is to think about how these apply to algorithms and pseudocode as opposed to a true programming language. If the students can figure these problems out on pen and paper – then ideally in the future, they could take a formal programming course and pick up the language intricacies speedily.

	• AAP-1.D.6 • AAP-3.A.4 • AAP-3.D.4 • AAP-2.O.4	• The use of lists allows multiple related items to be treated as a single value. Lists are referred to by different names, such as array, depending on the programming language. • A procedure call interrupts the sequential execution of statements, causing the program to execute the statements within the procedure before continuing. Once the last statement in the procedure (or a return statement) has executed, flow of control is returned to the point immediately following where the procedure was called. • Application program interfaces (APIs) are specifications for how the procedures in a library behave and can be used. • Knowledge of existing algorithms that use iteration can help in constructing new algorithms. Some examples of existing algorithms that are often used with lists include: ○ determining a minimum or maximum value in a list ○ computing a sum or average of a list of numbers
<u>Unit 8: AP Exam Project/ Exam Review/Post Exam</u>	n/a	Students will be provided 12 hours of class time to complete a culminating assignment equal to 30% of their AP Exam Score. This involves investigating computing innovations and algorithms – and creating a small program to demonstrate these ideas. After the assignment is completed and submitted – time in class will be spent focusing on reviewing for the written portion of the AP Exam. Once the exam has been completed – we will cover topics selected by the students and end with an in class final project.
<u>Additional Notes</u>	Topics order subject to change based on class needs. Students who are passionate about the course are encouraged to take Computer Science I and II if they have not done so already.	