

Computer Science Principles

Period 0 (8:10–9:00) • 2024-2025
Mr. Holzer • tholzer@ahsmail.com

“Every discovery in science and art, that is really true and useful to mankind has been given by direct revelation from God, though but few acknowledge it. It has been given with a view to prepare the way for the ultimate triumph of truth, and the redemption of the earth from the power of sin and Satan. We should take advantage of all these great discoveries, the accumulated wisdom of ages, and give to our children the benefit of every branch of useful knowledge, to prepare them to step forward and efficiently do their part in the great work.”

—Brigham Young, 1862

Course Description

Computer Science Principles is a course for all 9th graders at American Heritage School – Salt Lake City. This hands-on, project-based, introductory course is designed to help you learn the fundamentals of computer programming, regardless of whether you have had previous programming experience. If you can tell the difference between a computer and a toaster, you have all the prior experience needed to succeed in this course.

Computers are powerful tools — arguably some of the most powerful that our race has ever invented. The purpose of this course is to put the full power of the computer within your grasp. By learning how to program a computer, you will gain the ability to wield the computer as a tool to accomplish whatever your life work entails, including blessing yourself, your family, and the world. Further, whether or not you become a professional computer programmer, this class is for you, because computer programming skills can be applied *to every field of work* in which you will be engaged. For these reasons, we believe that *everyone should learn computer science* and thereby master the use of the computer as the powerful tool that it is — to bless yourself, your family, and others, and to build the kingdom of God on the earth.

In this course, you can expect to do the following four things:

1. Participate in readings, discussions, and activities that will give you a broad understanding of computing, including the legal, ethical, social, safety, privacy, and spiritual implications of computers and computer technologies. In a world that contains more computers than human beings and that requires any given person to interact with dozens of computers every single day, these are essential topics for us to discuss.
2. Develop core computer programming skills using the programming language Python. Not only do these skills allow you to program computers, but they will also help you think analytically — a skill that can be applied to much of our life’s decision making.
3. Learn to use specialized programming tools that will make computer programming a useful skill to you, regardless of what your life’s mission entails. Here is a sampling of some of the things you will learn to do.
 - a. Analyze and visualize data — Data is present in nearly every field of study, and computers are especially good at processing large sets of data and displaying them in useful ways.
 - b. Create graphical user interfaces (GUIs) — This skill will enable you to create computer programs that others can easily use and interact with.

- c. Create computer graphics — Being able to draw and animate windows, shapes, and pictures on a computer screen opens the door to basic web development, app development, game development, and related fields.
 - d. Create simulations — Simulating real-world situations, especially those that are difficult or dangerous to study directly, is another great strength of computers, and is applicable to nearly all fields.
 - e. Understand the basics of artificial intelligence and machine learning — Artificial intelligence is already a powerful technology, and it will likely continue to become even more powerful. Therefore, understanding the function, strengths, and limitations of this technology will simply become more and more important. Though the inner workings of machine learning can be very complex, our class will engage in projects and experiments that illuminate the basics. For instance, you will be given the opportunity to train a few machine learning models in class.
4. Use all of the skills you have learned throughout the year to create a final computer program in an area that interests you and that will bless others.

Which of the four points above are you most interested in? (This will help Mr. Holzer understand what you want to get out of this class.)

Does anything in the Course Description worry or concern you? (Again, this will help Mr. Holzer.)

Assignments and Exams

All assignments are due on the set due date. All late work will incur a 20% late penalty. You should plan ahead, knowing that more often than not, newly coded computer programs will not initially function correctly, requiring additional time for debugging and testing.

A paper copy of all assignments, readings, and project instructions will be handed out in class; they will also be posted to Gradelink, insofar as that is allowable by copyright law. If you lose or damage the printed copy that is provided to you, you will be responsible to print another copy from Gradelink.

All work for this course will fall into one of the following three categories. Each category has its own submission rules and due dates.

1. Weekly Homework Assignments

Due each Thursday

On most Thursdays, you will receive a homework assignment for the next week. Your homework is to be completed and turned in by the next Thursday in class. You are encouraged to begin working on your homework assignment early, so that you can ask questions and get help in class if you get stuck. Please note the following:

- **No late homework assignments will be accepted.** Exceptions will only be made in the case of an emergency, and an email from your parent will be required explaining the circumstances that prevented you from turning in your assignment on time.
- Homework must be completed by hand and turned in in class. Electronic submissions will only be accepted in exceptional circumstances.

2. Programming Assignments

Due at the end of each unit

For each programming unit, you will complete the following assignments and exams. All will be due by midnight on the last day of the unit. Each unit's due date will be clearly communicated to you as it approaches. It is highly recommended that you do not procrastinate.

1. In-Class Checkpoints: Checkpoints are short programming exercises that will be done in class. Once you have completed the exercise, you will submit a screenshot of the functioning code in our Microsoft Teams class. Each checkpoint is worth 4 points, with an additional extra credit point available on some checkpoints for completing an optional “stretch goal”. The checkpoints will be graded as detailed below. If you get a less-than-perfect grade on your checkpoint, you are welcome to resubmit it as many times as you like before the due date.

5 – Flawless + stretch goal
4 – Flawless
3 – Functional, but incomplete
2 – Functional, but incomplete
1 – Not functional
2. Programming Projects: We learn best by doing. For this reason, every unit will involve the completion of a larger programming project that applies the skills you will have learned in that unit. Time in class will be provided for you to work on these projects and get help, but you should also expect to spend some time outside of class for each project. Completed projects will be submitted in our Microsoft Teams class.
3. Python Reference Guide: As Python programming concepts are presented and practiced in class, you will be expected to capture them in what will become your own personal reference guide. This will be a major focus of our class: nearly every day you will make, annotate, and refine these notes. This ever-growing reference guide will be the primary source you will consult when working on programming projects. At the end of each programming unit, your reference guide will be turned in for grading through our Microsoft Teams class.
4. Unit Exam: On the last day of each unit (except Unit 7), you will take an exam covering all of the material from that unit. Your main study resource will be the checkpoints, programming projects, and homework assignments from that unit. Additionally, you will be allowed to use your Python Reference Guide on the exam. (Note that in order to use your Python Reference Guide on the unit exam, you will need to submit it one day early: by the midnight before the exam. It will be checked for unallowed content, and if it is acceptable, it will be printed and made available for you at the start of the exam.)

3. Term Assignments

Due at the end of the term:

- Term 1: Wednesday, October 16, 2024
 - Term 2: Thursday, December 19, 2024
 - Term 3: Thursday, March 6, 2024 (before Experiential Learning, which is the last week of Term 3)
 - Term 4: Friday, May 16, 2025 (one week before the end of Term 4)
1. Memorization — Each term, a scripture or prophetic quote will be provided for memorization. We will review the quote periodically in class to aid in the memorization process, but it will be your responsibility to memorize the quote, complete the accompanying reflection, and pass it off. You may pass off your memorization during any class work time; I will simply ask you to repeat the quote from memory and turn in your reflection.
 2. Class Notebook — Throughout the year, you will be expected to maintain a class notebook containing your homework assignments, your annotated readings, your Python Reference Guide, the final source code and worksheet for each of your projects, and reflections on what you are learning. Your best work is expected, including good handwriting and neat organization. Your notebooks will be inspected and graded at the end of each term.
 3. Research Projects — A handful of other readings and research projects will be given throughout the year. Specific instructions for each project will be provided in class.

Grade Scale

Grades will be assigned according to the standard American Heritage breakdown, as follows:

A = 100-93%	A- = 92-90%	B+ = 89-87%
B = 86-84%	B- = 83-80%	C+ = 79-77%
C = 76-74%	C- = 73-70%	D+ = 69-67%
D = 66-64%	D- = 63-60%	
F = 59-0%		
IN = Incomplete		

Course Materials

Articles — The following articles will be read and discussed in class, either in full or in part.

- "A century of living': In celebration of his 99th birthday, President Nelson reflects on innovation, growth, political breakthroughs" (Church News, 10 September 2023).
- "I, Chatbot" by Sara Smith Atwood and ChatGPT (Y Magazine, Spring 2023)

Talks — The following talks from leaders of The Church of Jesus Christ of Latter-day Saints will be read and discussed in class, either in full or in part.

- "Things As They Really Are" by Elder David A. Bednar (CES Fireside, 3 May 2009)
- "To Sweep the Earth as with a Flood" by Elder David A. Bednar (BYU Campus Education Week, 19 August 2014)

Books — Selected portions of the following books will be read.

- *Blown to Bits: Your Life, Liberty, and Happiness After the Digital Explosion, Second Edition*, by Abelson, Ledeen, Lewis, and Seltzer (Pearson Education, Inc., 2021)
- *Clean Code: A Handbook of Agile Software Craftsmanship*, by Robert C. Martin (Pearson Education, Inc., 2008)
- *Debugging: The 9 Indispensable Rules for Finding Even the Most Elusive Software and Hardware Problems*, by David J. Agans (AMACOM, 2002)

Any other materials that become part of the required reading list will be communicated clearly to you and to your parents.

Class Policies

Respect for People — The highest standard of respect for others is expected in our class: for your classmates, for all faculty and staff, and for any volunteers who may assist us in any way. I promise to treat you with the same level of respect, for I view the opportunity to interact with you in class as a profound privilege.

On-Time Attendance — This class begins at 8:10 a.m. This is early, but we will nevertheless start on time every day. You are expected to be in your seat at 8:10 a.m., with a computer on your desk (if we will be using a computer that day), ready to begin. Please plan your rides to school accordingly. Being tardy once or twice during the year will be excused when things are out of your control, but regular patterns of tardiness will be noted and addressed by Mr. Holzer. If you do come into class late, you are expected to enter silently and move straight to your desk without drawing attention to yourself or disrupting anyone.

Technology Use

- Computers — You will have almost daily access to the school-owned laptops. You will be expected to use them respectfully and only for the purposes explicitly outlined in class. After a single warning, any disregard for this policy will result in loss of computer privileges for the remainder of the class period. This same policy also applies to personal computers: you may use your own personal computer in class, but only when other students are using the school computers and only for the purposes explicitly outlined in class.

- Phones — In accordance with the policies of American Heritage, phone use in class is not permitted.

Cleanliness — We all need to work together to keep our classroom clean. It is expected that everyone will help to clean up at the end of each class, including putting school computers neatly and respectfully back into the computer cart. Once you have finished cleaning up your own work area, please look for opportunities to help others clean up as well.

Group Work — Collaboration is an essential skill in the field of computer science, and you will nearly always be allowed to collaborate with your peers when completing class projects. However, you must complete each project yourself; you are not allowed to turn in anyone else's work. If it is suspected that you have copied someone else's work, we will have a frank conversation that will likely result in you needing to redo your work. Repeated violations of this policy will be escalated to school administration.

Artificial Intelligence Use — Artificial intelligence is a powerful tool and has many legitimate and powerful uses, some of which will be discussed and explored in this course. However, because this course is focused on developing your own personal skills of logic, reasoning, and computer programming, you will not be allowed to use generative AI tools of any kind for any of the work done in this course, including debugging non-functional computer code. American Heritage aims to offer a unique style of education in which both the heart and mind are educated, and we believe that developing the time-honored skills of thinking for yourself, reasoning, and writing are some of the best methods of achieving this kind of transformative learning. Relying on AI to do your thinking, writing, coding, and debugging will prevent you from learning the skills that you can only learn by doing it yourself. We believe that this approach will prove a great advantage to you, including if you finish this course and decide to begin using AI to do some of the work that you did yourself while completing this course. Therefore, we ask for your trust as we insist that this policy be kept by all students in this course.

If you have concerns about this policy, please talk with me; I will be happy to discuss this with anyone. If it is suspected that you have used an AI tool to do any of your thinking, writing, or programming, we will have frank conversations that will likely result in you needing to redo your work. Repeated violations of this policy will be escalated to school administration.

Do you anticipate having any problems keeping these policies? Do you have any special circumstances or needs Mr. Holzer should be aware of?

I agree to abide by these policies. (Sign your name to signify your agreement.)

Questions

Do I need to be able to program at home?

Yes. Time will be provided in class for you to work on your programming projects, but you will usually need to spend some additional time at home to complete them. This will require that you install the latest stable version of Python and a programming text editor on a computer which you can access outside of class. If this poses any problems for you, please talk with me; I will be happy to make accommodations for you.

The installer for the latest version of Python (currently Python 3.11.5) can be downloaded from <https://www.python.org/downloads/>. You are allowed to use any programming text editor you wish, but I would strongly suggest that you use Visual Studio Code, as this is the editor we will use in class. You can download Visual Studio Code from <https://code.visualstudio.com/>. A step-by-step guide for installing both Python and Visual Studio Code will be provided to you during the first week of class. If you run into any problems with installation, please reach out to me early so we can get them resolved.

Why Python?

Many different programming languages could have been selected for this course, each with different strengths and weaknesses. JavaScript, HTML, and CSS are heavily used in web development; Java is used to write cross-platform applications and mobile apps; and C and C++ are standard languages often used for programming embedded devices. Any of these languages could potentially be used to introduce you to computer programming. However, after careful consideration, Python has been selected as the programming language for this course because of its versatility, universal support, and ease of use.

This course is designed to be taken by all students, to equip them with enough programming knowledge that they could use computer programming in any other area of interest throughout their lives. After taking this class, you may choose not to learn more programming. With that in mind, I asked myself the following question: If a student were to learn only one programming language, which would be the easiest for them to learn and provide the greatest long-term usefulness to them? With these criteria in place, Python seemed to be the best choice.

Using Python will get you creating meaningful programs quickly, without getting lost in some of the more detailed complexity inherent in other languages. While those details are certainly important for those who choose to continue learning computer programming and want more control, I don't want students to give up due to unnecessary complexity right at the beginning of their programming experience. To employ the use of an analogy, a music teacher generally would not choose to teach a toddler how to play the organ; instead, the toddler would start by learning the piano, where he does not need to bother with the complexity of pedals and stops and multiple registers. In much the same way, Python does not require the immediate understanding of many low-level concepts that can be difficult for beginners to grasp.

But please don't misunderstand me: Python is not a play language! To extend the use of our music analogy a little further, just because the piano is usually the first instrument keyboard musicians learn does not mean that it is an inferior instrument. It is not a kazoo or a plastic recorder! It is a real instrument and can produce incredible music — and for many musicians, it is their lifelong instrument of choice. Likewise, Python is not an overly simplified programming language useful only for beginners (like Scratch and other drag-and-drop languages): it is a real programming language, used by real computer programmers to create real (and really cool!) things. Not convinced? Did you know that YouTube's backend was written mostly in Python? Same with Netflix and Pinterest. Google's and Facebook's backends also use some Python. Python is widely used by scientists and engineers for data analysis and visualization, and the quickly growing field of artificial intelligence and machine learning has adopted Python as its de facto standard. In fact, for two decades, Python has been ranked as one of the most popular and widely used programming languages in the world, and by all indications, this will not change any time soon.

Will I be encouraged to use artificial intelligence tools?

You will not be encouraged to personally use artificial intelligence tools developed by others. However, as part of our classroom discussions on this important topic, demonstrations of certain uses of artificial intelligence will be given in class, including the use of large-language models, like ChatGPT, and possibly some AI image generators. The purpose of these demonstrations is to facilitate the analysis of what artificial intelligence is, what its current capabilities and limitations are, and what world views and attitudes it reflects. Additionally, you will have the opportunity to employ some simple tools to train a machine learning model, with the purpose of understanding how machine learning works.

Just like computers and the Internet, AI is a powerful tool. As is true of all powerful tools, AI can be used for both good and bad. Under no circumstances should you use ChatGPT or other artificial intelligence tools to complete your assignments for you. Doing so in this course will be considered dishonest and will hurt you in the long run, for it will rob you of the very learning, thinking, writing, and feeling opportunities that you have come to American Heritage to obtain. There are, however, appropriate ways to use AI, which we will discuss as part of this course.

Will I be encouraged to use search engines to find information online?

Yes. The use of technology is generally discouraged during class time at the Salt Lake Campus of American Heritage School, as it often distracts from the opportunity to interact with other people — both your teachers and fellow students. However, technology and the Internet are such huge parts of our society and carry such great potential for good that *in this particular class*, you will not only be encouraged to use these tools, but also taught appropriate ways to use them. At no time, however, will “surfing the Internet” be tolerated. Before any Internet research is performed, we will carefully discuss appropriate Internet use, how to identify reputable sources of information, and which search engines and websites are most likely to provide wholesome, accurate, legal

information and images. During class periods in which we conduct Internet research, your computer screen will at all times be visible to me and other class members.

The Internet has become the de facto repository for official programming documentation. Finding information, examples, and tutorials to teach yourself new programming concepts is a very helpful skill when programming a computer. In the initial stages of this course, I will aim to give you as much information as I can, thus minimizing the amount of Internet searching that must be done. However, during the final project, you will likely need to obtain additional information from the Internet. You will be pointed to good sources, so that in most cases, you will not need to resort to conducting broad Internet searches. But in some cases, conducting an Internet search will be the most efficient way for you to obtain the information you need. If you or your parents are concerned, please contact me. I will be very happy to work with you and to provide alternative activities or resources.

Programming Units

Unit 1: Basic Python Programs

Discussion Topics

- What are Computers?
- Computers as Tools
- Computers as a God-given Endowment for the Latter-day Work
- Brief History of Computers
- What are Computer Programs?

Python Programming Modules

Basic Printing, Variables, Operators, and User Input

1.0 – Setup Environment, Hello World!

1.1 – Basic Printing and Comments

1.2 – More Printing

1.3 – Numbers and Math Operators

1.4 – More Operators

1.5 – Variables

1.6 – Multiple Lines and Escape Characters

1.7 – Getting User Input

1.8 – Getting Numerical User Input

1.9 – Printing with Format Strings

Programming Project: Fill-In Story

Students will create a Python program that generates a “fill-in” story (also called a “mad lib” story). The program will begin by asking the user to enter words of different types. These words and several randomly generated numbers will be inserted into a student-written story, which will be printed to the terminal.

Unit 2: Decision-Making

Discussion Topics

- Control Flow in Programs
- Decision-Making in Computer Programs and in Life

Python Programming Modules

Decision-Making

2.1 – If-Else Statements

2.2 – Comparison Conditions and Elif Statements

2.3 – Testing for Inequality and Multiple Conditions

2.4 – Checking User Input

2.5 – The Or and Not Operators

2.6 – Nested Conditionals

Programming Project: Personality Profile

Students will write a Python program that tells the user his or her personality type, based on the answers to five questions about his or her personality. There are 16 different personality types, all of which are different combinations of the five main personality traits. This project will require extensive use of if-else statements.

Unit 3: Data Representation and Organization

Discussion Topics

- Binary Numbers
- Data Types
- Organizing Data
- Data Abstraction

Python Programming Modules

Data Representation

3.1 – Converting Between Data Types

Lists

3.2 – List Creation and Indexing

3.3 – Manipulating Lists

3.4 – List Slicing

Dictionaries

3.5 – Dictionary Creation and Indexing

3.6 – Manipulating Dictionaries

Class Activity: Binary Messages

Students will write an encoded message to a classmate by converting English characters into binary bits using an ASCII table and their knowledge of the binary representation of decimal numbers.

Programming Project: Top Ten List

Students will write a Python program that defines a “Top Ten” list of their choice and displays it on the screen.

Unit 4: Iteration and Algorithms

Discussion Topics

- Repetition and Iteration
- Algorithms
 - What are Algorithms?
 - Building Blocks of Algorithms: Sequences, Decisions, Iterations
 - Process of Algorithm Development

Python Programming Modules

Loops

4.1 – Looping Over Ranges

4.2 – Looping Over Lists

4.3 – While Loops

4.4 – Creating Lists in Loops

4.5 – Combining Decision-Making and Iteration

4.6 – Rounding Numbers and More with Lists

Programming Project: Virtual Waiter

Students will write a Python program that displays a restaurant menu, takes an order from a customer, totals up the price for the order, and then prints out a receipt for the customer. Completing this task will require students to organize menu items and customer orders using lists, to iterate through those lists, to loop an unspecified number of

times to allow the customer to order as many items as they want, and to practice the skills of algorithm development that are taught in class.

Class Activity: Max Number Algorithm

Students will work in pairs to design an algorithm that identifies the maximum number in an unsorted list of numbers. They will diagram the algorithm, specifically identifying the steps, decisions, and iterations involved. They will then team up with another group and compare the two algorithms, considering whether each algorithm will yield the same result and which algorithm will take the least number of steps to complete. Each group will then code their algorithm in Python. When the algorithms are completed, groups will give their code to another group for testing. Students will develop a list of test cases to ensure that the algorithm is working.

Unit 5: Abstraction and Modularity

Discussion Topics

- Abstraction: A Key Concept in Computer Science
- The Advantages of Modularity

Python Programming Modules

Functions

- 5.1 – Defining and Naming a Function
- 5.2 – Input Arguments
- 5.3 – Return Values
- 5.4 – Practice Calling Functions
- 5.5 – Functions with Boolean Return Values
- 5.6 – Functions with Lists

Programming Project: Fact Game

Students will collaborate in groups of two to develop a program that tests a user's knowledge of a certain set of facts (a virtual flash-card game). Students will develop a number of functions to use in their program, including functions that display a question and receive the user's answer, check the user's answer, update the user's score, add questions the user got wrong to another list for further quizzing, and print the user's final score.

Unit 6: Inputs and Outputs

Discussion Topics

- Identifying Inputs and Outputs
- Improving the User Experience

Python Programming Modules

File Operations

- 5.1 – Reading a File
- 5.2 – Searching a File
- 5.3 – Writing to a File
- 5.4 – Appending to a File

Graphical User Interfaces (GUIs)

- 5.5 – Creating a Tkinter GUI
- 5.6 – Using a GUI to Make an Interactive Program
- 5.7 – User Input in a GUI
- 5.8 – GUI Geometry Managers

Programming Project: Scripture Search

Students will write a program that reads in the entire text of a book of scripture, searches it for user-specified words or phrases, and then displays the results (including the number of times that word appears in the text). A simple

GUI will be developed, which will allow the user to type in their desired search phrase, click a button to initiate the search, see the results on the screen, and click a button to initiate an export of the results to a file.

Unit 7: Final Project

Programming Project: Final Project

Students will each create a final project: a significant Python program that intersects with one of the student's passions or areas of interest and could make a difference for good by positively impacting the student, the student's family, or others. Each student will create his or her own project, but students will be paired with a collaboration partner; over the course of the project, collaboration partners will work together on identifying key steps in the algorithms of the two programs, generating a list of test cases, and solving problems that arise during the development process. All projects will be demonstrated to the class at the end of the school year.

Discussion Topics

The following topics will be discussed in class at different times throughout the year, interspersed with the Programming Units above.

Topic 1: The Impact of Computing on Our World

Discussion Topics

- Reading and Class Discussion: *Blown to Bits*, Chapter 1: Digital Explosion
- Reading and Class Discussion: "A century of living? In celebration of his 99th birthday, President Nelson reflects on innovation, growth, political breakthroughs" (Church News, 10 September 2023).

Research Project: Impact of Computing Interview and Essay

Students will conduct an interview with a family member or neighbor who was born before 1971 and thus has witnessed firsthand the transformations brought about by the digital revolution. They will then use what they have learned in class and in the interview to write an essay responding to the following prompt on how computing has impacted individuals, families, and the world, and whether they believe that the positive effects of computing outweigh the negative effects.

Topic 2: The Internet

Discussion Topics

- What is the Internet?
- How the Internet Works
- Moderation of Information on the Internet
 - Reading: *Blown to Bits*, Chapter 4: Gatekeepers
- How does an increased understanding of the Internet help us use it effectively and safely?

Class Activities: Transmitting Messages through the Internet

Students will develop an intuitive understanding of how information is passed through the Internet by engaging in a number of classroom simulation activities, in which students pass index cards from person to person in the same way that computers connected to the Internet pass data packets.

Topic 3: Privacy and Safety in a Digital World

Discussion Topics

- Advantages and Disadvantages of Technology with Regard to Privacy and Safety
 - Reading and Class Discussion: *Blown to Bits*, Chapter 2: Naked in the Sunlight
- Protecting Individual and Family Privacy in a Digital World

Class Activity: Legislative Roleplay

Students will work in groups to discuss potential policies, laws, or Constitutional amendments that could be enacted to solve some of the privacy and safety problems posed by digital technology. Each group will present their ideas to the class and the merits and drawbacks of these ideas will be discussed.

Topic 4: Legal and Ethical Concerns of Computing

Discussion Topics

- Legal and Ethical Issues Arising from Technology
 - Reading and Class Discussion: *Blown to Bits*, Chapter 6: Balance Toppled
 - Reading and Class Discussion: *Blown to Bits*, Chapter 7: You Can't Say That on the Internet
- Intellectual Property
- Ethical Behavior Online — Including the Spiritual Implications of Unethical Behavior

Research Project: How-To Guide for the Digital Age

Pretending that a hypothetical family member had been asleep for the last 100 years and had just awoken in the digital age, students will create a written guide for this family member to help them navigate the legal, ethical, safety, and privacy concerns of our time. Bonus points will be given for creativity, appropriate humor, and accompanying illustrations.

Topic 5: Internet Research

Discussion Topics

- Types of Sources: Primary, Secondary, and Tertiary
- Website Categories
- Evaluating Internet Sources for Reliability
- How Search Engines Work
- The Dos and Don'ts of Google, YouTube, and Wikipedia
- Other Helpful Internet Sources

Topic 6: Artificial Intelligence and Machine Learning

Discussion Topics

- What is Artificial Intelligence?
- What is Machine Learning?
- Human versus Computer: "Thinking"
- Strengths and Weaknesses of AI
- ChatGPT and Artificial Intelligence Today
- Reading and Class Discussion: *Blown to Bits*, Chapter 9: The Next Frontier

Class Project: Train a Machine Learning Model

In class, students will be coached through the process of training a machine learning model in Python.

Topic 7: Using Computers for Good

Discussion Topics

- Computers and a God-Given Tool for the Gathering of Israel
- Recognizing Truth in an Age of Artificial Intelligence and Virtual Reality