General Instructions

1. For each page, synthesize the reading and research by writing in complete sentences in essay format.
   a. Use the green instructions in the notes below each page to focus your research and writing.
   b. Write about each question, prompt, or process provided in those notes.
   c. Write a full page of text with lots of detail (about 250 words).
      i. Don’t generalize so much that your writing is devoid of detail.
      ii. Don’t repeat yourself.
   d. Cite each source by adding a hyperlink in the Title of the Article or law.
   e. Do not change the template:
      i. Text must be 14 point Lato left-justified type.
      ii. Refrain from adding extra margins or double spacing.
      iii. Do not bullet the paragraphs. Write in essay format only.
   f. Add additional pages if you need more room.

2. Add all sources to the Bibliography page.
   a. Include author, year, title, publisher, and URL.
   b. Number or bullet them using the list button.

When in doubt, write to the instructor for clarification using the Canvas Inbox.
When I graduate, I would like to work as a software engineer for an exciting tech company like Google. According to a job listing on Google’s [Careers search engine](https://careers.google.com), a Google software engineer must have a minimum of a bachelor’s degree, or preferably a master’s degree, in computer science or a related field in order to be considered for the job. There are no certificates explicitly stated as requirements for the job, but having certain certifications and exam results can make a resume more appealing. Google software engineers definitely have to adhere to codes of ethics. Google is one of the world’s largest technology companies and runs one of the world’s largest email services. As such, the company has a vast amount of private data stored that needs to be protected. Software engineers have an ethical duty to keep software secure so that data isn’t leaked outside the company. The job list specifically notes that Google makes a diverse line of products for diverse users. The company values diversity and engineers must be proactive about avoiding discrimination. That means making a products for all members of society while making those products available to as many people as possible. Software engineers also have a duty to ensure honest, non-deceptive practices in their work. Almost all of an applicant’s credibility in the field of software engineering comes from experience. That is why it is so important to get practical experience through internships. Most internships only require college coursework and a demonstration basic understanding. Because software can be developed by anyone at anytime, it is also important for applicants to develop a portfolio of their work. This can consist of projects done for employers or personal projects done for personal enrichment. In software engineering, practical experience is as important, if not more important, than an education alone.
As a software engineer, I will likely be working with users’ private data. As such, I will need to be extremely cautious about how I handle it. Current software infrastructure is designed to avoid data leaks and breaches, but there are still vulnerabilities and adding, removing, or altering code can introduce new vulnerabilities. According to Data Breach 101: Top 5 Reasons it Happens (Data Breach 101), outdated software is one of the leading causes of data breaches. Whenever software is changed to patch a vulnerability, it is likely that there are still many unknown vulnerabilities that still exist and that new vulnerabilities may have been introduced. This is because human error is always a factor in software development. If I try to patch a vulnerability or add a new feature to a large system, it is nearly impossible for me to consider all of the edge cases those changes introduce. Other leading causes of data breaches include malware, insider misuse, and physical theft of a data storage device. The issue of data breaches relates to the ACM Code of Ethics clause 1.7 Honor Confidentiality. The clause states that I should protect the confidential information entrusted to me by a user. The only exception to this rule is that confidential data can be disclosed to the proper authorities if it is evidence of the violation of a law.

One way I would avoid future vulnerabilities is to keep software updated. Leaving software unpatched for an extended period of time gives bad actors a chance to build exploits. Even if all the exploits can’t be patched, maintaining an ever-changing system will reduce the time frame for exploits to be effective. I would also actively be conscientious about my code. I would raise awareness to reduce human error. The third thing I would do is be careful about how data is transferred. I could reduce physical theft by storing and transporting physical assets securely.
Many people fear that automation will soon take all the low-wage jobs. While automation is only beginning to spread across low-wage jobs, such as the fast-food industry, I don’t think it will be long before human positions start disappearing. Low-wage jobs are generally repetitive tasks which makes them ideal candidates for automation. With the recent advancements in robotics and computing technology, most jobs can be easily automated by a robot that workers more efficiently than a person while requiring no wage. In his article *How Long Will It Take For Your Job to be Automated*, Richard Gray discusses some proposed time frames for when certain jobs may be automated. According to a graphic in his article, robots are predicted to be folding laundry, driving trucks, and acting as a retail salesperson by about 2030. I would estimate the automated truck driver as coming sooner rather than later, but the general idea is there. I’m concerned about the loss of jobs, but also the risk of failure. It is not particularly uncommon for machines to malfunction or be hijacked by a bad actor. I’m concerned that an automated system that becomes centrally integrated into society could fail. This relates to the ACM Code of Ethics clause 3.7 Recognize and Take Special Care of Systems That Become Integrated Into the Infrastructure of Society.

One way I can address the potential job crisis is to be cautious about promoting and implementing automation solutions. I should ensure that the solutions I develop will not displace a critical number of workers without creating new job opportunities. I could also take steps to maintain the stability of automation solutions currently in use. This will help avoid a catastrophe related to the malfunction of critical machinery. A third thing I could do is try to create jobs. Automation solutions need maintenance so I could create jobs for displaced workers maintaining systems.
Big data is the term used to describe the massive amount of data stored in databases. One of the biggest questions in computing and data science is how to most effectively use that data to make the world better. According to Why Predictive Analytics and Microtargeting is a Game-Changer For Marketers, advancements in data analytics and artificial intelligence have made it possible to analyze a particular user’s data—including search history, purchase history, location, and identifying information—to target them with ads predicted to relate to that user’s interests. Many people disagree with micro-targeted advertisements, but for different reasons. Some people find it annoying to be bombarded with advertisements for something you just purchased, others don’t like the idea of major tech companies having such a large stockpile of information about them. I don’t mind targeted advertising very much because it reduces the number of completely irrelevant advertisements that I see. I do, however, believe individuals should have a choice about whether they are being micro-targeted with advertisements. This most closely relates to the ACM Code of Ethics clause 1.6 Respect Privacy because I would consider it a breach of privacy to collect and apply data about a user’s computer usage without that user’s knowledge and consent.

One way I would address this issue is by always informing a user of how their data will be collected and used. Depending on the application, I might also give the user the choice to opt-in or opt-out of the data collection. Similarly, I would give the user the option to see targeted advertisements or random advertisements. I think the key in this dilemma is less about methods than it is about user choice. I would advocate against using targeted advertisements to push an agenda and would take precautions to avoid leaking data to other companies that would.
According to a Stanford report titled *The Digital Divide*, the digital divide is the expanding gap in access to technology between the underprivileged member of society and the wealthier upper class, middle class, and young Americans. The digital divide also applies to those who don’t understand digital media, like the elderly. Although the total amount of digital technology increases with the advancement of technology, the digital divide also becomes larger. Technology has become so ingrained in society that access to a smart device (i.e. smartphone, tablet, computer, etc.) is almost necessary to be a functioning member. For example, I have noticed that there are almost no physical job applications anymore. Nearly all job applications are digital, meaning a smart device is necessary for completing and submitting the form. This makes applying difficult for an underprivileged member of society only allows the digital divide to grow. This is one example, but the list can go on. This idea relates closely to the ACM Code of Ethics clause 1.4 Be Fair and Take Action Not to Discriminate. The clause preaches equality, saying that I, as a computing professional, should foster fair access for all members of society, including underrepresented groups.

One way I would address this dilemma is by making smart devices more available. As a software engineer, I will be limited in the ways that I can do this, but a good start would be building a low-cost smart device that underprivileged groups could use. Another option would be to promote the availability of accessible media. This would mean advocating for media that doesn’t require a smart device to access. To address the issue of job applications, employers or employment agencies could make computers available for unemployed persons to use to apply to jobs. I could also build a low-cost system designed to educate the user about the use of digital systems.
Copyright laws apply to many ideas and software is no exception. According to Copyright Your Software—Why Bother, there is an automatic copyright that applies to software without the copyright needing to be registered. However, registering is still recommended for total protection. There are a few ways software can be copyrighted in my profession. The most obvious way is by copying a piece of software. If I tried to distribute Adobe Photoshop for my own profit, that would be copyright infringement. Another form of infringement more specific to my field has to do with the source code of the software. A copyright can be infringed upon even in a single script is copied if the creator of that script can show that it is proprietary source code. An example infringement is with the Skyrim Together mod for The Elder Scrolls V: Skyrim. The developers of the mod were accused of using a slightly modified script taken from the Skyrim Script Extender mod created by another team. This relates to the ACM Code of Ethics clause 1.5 Respect The Work Required to Produce New Ideas, inventions, creative works, and computing artifacts.

The main thing I would do to avoid copyright infringement is not use any proprietary code in my software that doesn't belong to me. The issue with Skyrim Together was that they were given permission to use the scripts but, after a falling out between the teams, permission to use the script was revoked. I would be careful not to use scripts like this without written permission to implement the code permanently. I would also copyright all of my own code to keep my intellectual property safe. A good way to get help without infringing on copyright is to use open source code and open source software. Open source is uncopyrighted, meaning that anything it contains is free for anyone to use and implement. To keep others from infringing on my copyrights, I would clearly label my work as proprietary.
I would have to blow the whistle if I learned my company or coworkers were infringing on software copyrights. The first reason is that I think of software as being similar to an invention. At first glance, a script looks like just a list of commands that anyone could have written, but an elegant solution to a complex problem can look deceptively simple. Writing an elegant and efficient script to a complex problem takes a long time. To copy and distribute another person’s code isn’t a victimless crime and the code copied likely isn’t a solution the perpetrator would have conceived of alone. Infringing on software copyright is undeservingly profiting from another software engineer’s hard work. The other reason I would blow the whistle is that the act is so clearly illegal. There are situations where a whistle could be blown, but the harm of the activity is too subjective for me to justify. For example, the whistle was blown on Cambridge Analytica because they were generating targeted advertisements based on data. However, much of this data was gathered legally and determining what a company can legally do with data on a website over which it has jurisdiction is a fuzzy subject. Copyright infringement, on the other hand, is explicitly illegal and it is relatively easy to determine if a script was copied into a project.

Before blowing the whistle, I would talk to my coworker or boss to see if there is some way to resolve the issue. If this didn’t work, I would start the whistleblowing process with the United States Department of Justice by reporting a computer, internet-related, or intellectual property crime. After reading the article referenced here, I see that the U.S. Department of Justice page referred me to my local FBI office, so I would contact them next. As a whistleblower, I would likely lose my job and delay or terminate the project using the scripts.
I considered most of these dilemmas from a Kantian deontological perspective. Kantian ethics argue that there are certain rules that cannot be broken and there are certain duties that must be followed. In the software dilemma, I pointed out that the most common causes of data breaches are unpatched software and theft stemming from human error. For this reason, I have a duty to ensure my software is constantly being updated and I’m taking due care not to be complacent when handling data. In the hardware dilemma, I noted that one of the biggest issues with automation is that it is coming too fast with too few replacement jobs for dislocated workers. Even though automation of the workforce is in the best interest of many people (i.e. consumers, manufacturers, computer scientists, electrical engineers, etc.), I believe I have a duty to promote automation at a manageable rate while considering how my solutions can create new jobs. My answers to this dilemma also relate to Utilitarianism because it considers the amount of good and bad introduced by the solution, but it is hard to tell which subset of the population is the largest and which is affected the most. My solutions for the advertising dilemma also related to Kantian ethics because I argued that I have a duty not to use data for ad targeting or pushing an agenda. My discussion of the digital divide especially relates to this because I explained my duty as a software engineer to create accessible, low-cost smart solutions to close the gap. It is hard to say that my copyright argument related to Kantian ethics other than by saying I had a duty to do the right thing. This solution relates more closely to the Fairness and Justice Approach because I argued all software engineers should be fair in their use of software. I will not copy another engineer’s work for my own profit and I expect that others will be as fair to me.
● “Devising an Ethical Framework.” People.oregonstate.edu, Oregon State University, July 2019, people.oregonstate.edu/~vanlondp/cs391/articles/ethical-framework.php.

“Software Engineer.” Google Careers, Google, careers.google.com/jobs/results/133713543317856966/.
