IRIE
International Review of Information Ethics Vol. 31 (08/2022)

Kevin Baum, Sarah Sterz

Ethics for Nerds

Abstract:

Informatics is the innovation driver of our time. From social media and artificial intelligence to autonomous cyber-physical systems: informatics-driven, digital products and services permeate our society in significant ways. Computer scientists, whether researchers or software developers, are shaping tomorrow's society. As a consequence, ethical, societal, and practical reasons demand that students of computer science and related subjects should receive at least a basic ethical education to be able to do justice to their ever-growing responsibilities and duties. Ethics for Nerds is an award-winning lecture that is being taught annually at Saarland University since 2016. The course has been continually updated and progressively improved over the years. In this paper, we share our experiences with and best practices for teaching the basics of ethics to students of computer science and offer advice on how to design a successful ethics course as part of a computer science study program.

Agenda:

Authors:

Kevin Baum

- Researcher, Lecturer, Ph. D. student, Department of Philosophy & Department of Computer Science, Saarland University, 66123 Saarbrücken, Germany. Email: kevin.baum@uni-saarland.de

Sarah Sterz

- Researcher, Lecturer, Ph. D. student, Department of Philosophy & Department of Computer Science, Saarland University, 66123 Saarbrücken, Germany. Email: sterz@depend.uni-saarland.de

ISSN 1614-1687

Introduction

Ethics for Nerds is an award-winning lecture that is taught annually at Saarland University since 2016 and that has received excellent student evaluations. The course has been continually modified and progressively improved over the years. In this paper, we share our experiences with and best practices for teaching the basics of ethics to students of computer science.

It stands to reason that students of computer science and related subjects should receive ethical education. But how can lecturers achieve that and what are the most common obstacles? This paper will be laying down best practices from over half a decade of teaching ethics to computer scientists and will offer advice on how to design a successful ethics course as part of a computer science study program. Based on various considerations, we first give a motivation of why a course like this is at least collectively advantageous, and plausibly also morally obligatory. Second, we briefly survey the course content and, third, explain the set-up of Ethics for Nerds. Fourth, we discuss challenges that such a course must face, and, finally, we present a list of best practices linking back to the goals and challenges of the course.

Why Ethics for Nerds?

Practical and applied philosophy provides us with at least three reasons to teach ethics to computer science students: firstly, doing so plausibly is morally obligatory according to consequentialism; secondly, it enables adequate ascription of responsibility; and, thirdly, it is demanded by professional ethics. These three reasons provide a sound justification for the assumption that courses like *Ethics for Nerds* should be common practice, and, at the same time, provide substantial guidance in how to suitably design such a course.

Consequentialism

It is, by now, undisputed that computer scientists, software designers, and programmers have enormous impact on the world. Moreover, design decisions in these fields often come with multifaceted challenges that reach far beyond merely technical issues. Instead, they also touch on various other dimensions like society, morals, or law. These challenges must be tackled as the danger that lies in ignoring them or underestimating them is far too great.

Assuming this, it is plausible that a course along the lines of *Ethics for Nerds* is morally imperative for consequentialist reasons: first, the course sensitizes students to existing and potential future problems, which increases the likelihood that they will recognize such problems in their professional lives; second, the course enables students to identify morally better options and correspondingly make better decisions; and third, *Ethics for Nerds* enables the sustained embedding of these skills. In this respect, *Ethics for Nerds* has moral value in terms of information gain, applying the general decision-theoretic definition of information value[1] to the moral-theoretic framework of consequentialism.[2] In other words, *Ethics for Nerds* has inherited, instrumental moral value.

[1] Howard, "Information Value Theory," IEEE Transactions on Systems Science and Cybernetics [Aug. 1966] 2 (1), pp. 22-26, available at https://ieeexplore.ieee.org/abstract/document/4082064.

[2] See Askell, "The Moral Value of Information," EA Global: Boston 2017. Science Center, Harvard University, 2 – 4 Jun. 2017, Harvard Science Center, Cambridge, USA. The talk is available at https://www.youtube.com/watch?v=o8rVscSHJT4 and a transcript is available at https://www.effectivealtruism.org/articles/the-moral-value-of-information-amanda-askell/.

Adequate responsibility ascription

The last years have brought widespread agreement that computer scientists should bear moral responsibility for some of their research and developments.[3] What is called the *epistemic condition* is a necessary requirement for responsibility, i.e., that, roughly, the responsible party has to be able to assess the consequences of their doing and can at least suspect that the outcome in question might come about as a result of that – otherwise they are not responsible.[4] This crucially hinges on adequate knowledge of the world and of relevant concepts. In practice, this does not only mean knowing what will happen if we perform a certain action, but also knowing how to value certain outcomes in terms of, e.g., morals. If a researcher, for example, cannot assess what societally devastating consequences their action can have because they were never educated on that, they might be responsible to a far lesser degree. Likewise, if a software developer performs a morally wrong action because they never learned to tell apart morally right from wrong actions in complex professional situations, then they might bear less responsibility for this action. These issues can, at least in part, be mitigated by proper moral education which is tailored to the needs of future computer science and informatics professionals and which helps them to bear the responsibility that they should bear.

Professional and institutionalized ethics

Even though professional codes of ethics and codes of conduct are heuristics for professionals who are not trained ethicists, they assume some degree of ethical and moral literacy. So, for adequately using them, a certain moral education in computer science is useful and in practice oftentimes needed. The preamble of *the ACM Code of Ethics and Professional Conduct*[5], for example, reads: "The Code is not an algorithm for solving ethical problems; rather it serves as a basis for ethical decision-making." Many principles in the code rely on the reader's capability to make independent moral judgements, e.g., in *Principle 2.2* where the code asks to "[m]aintain high standards of professional competence, conduct, and ethical practice", or in *Principle 2.3* where it says that "[c]omputing professionals must abide by these rules [e.g., laws and regulations] unless there is a compelling ethical justification to do otherwise" and that "[r]ules that are judged unethical should be challenged". The *Software Engineering Code of Ethics and Professional Practice* by the *ACM/IEEE-CS Joint Task Force on Software Engineering Ethics and Professional Practices*[6] has a similar preamble: "The Code is not a simple ethical algorithm that generates ethical decisions. In some situations, standards may be in tension with each other or with standards from other sources. These situations require the software engineer to use ethical judgment to act in a manner which is most consistent with the spirit of the Code of Ethics and Professional Practice, given the circumstances." This code, too, assumes a proper judgement of moral issues, e.g., in *Principle 2.09* where it says to "[p]romote no interest adverse to their employer or client, unless a higher ethical concern is being compromised". Even though the *IEEE Code of Ethics*[7] takes a less explicit stance on pre-existing moral skills, it, too, has principles that call for a certain background knowledge, e.g., *Principle I.1* that asks for compliance with "ethical design".

[3] Gotterbarn, "Informatics and professional responsibility," *Science and Engineering Ethics* [2001] 7, pp.221–230; Noorman, "Computing and Moral Responsibility" in: Edward N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy* (Spring 2020), available at: https://plato.stanford.edu/archives/spr2020/entries/computing-responsibility/.

[4] See Talbert, "Moral Responsibility" in: Edward N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy* (Winter 2019), available at https://plato.stanford.edu/archives/win2019/entries/moral-responsibility/; for more details on the epistemic condition in context of responsibility in human-computer interaction also see Baum et al., "From Responsibility to Reason-Giving Explainable Artificial Intelligence," *Philosophy & Technology,* forthcoming.

[5] ACM Code of Ethics and Professional Conduct [2018], available at https://www.acm.org/code-of-ethics

[6] ACM/IEEE-CS Software Engineering Code [1997], available at https://ethics.acm.org/code-of-ethics/software-engineering-code/

[7] IEEE Code of Ethics [2020], available at https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-code-of-ethics.pdf

The Course

From the motivation in Section 1, we can take away three main goals that an ethics course for students of computer science should work towards:

1. The students are able to recognize potential moral problems of their profession and their work.
2. The students have the necessary tools to solve these problems or know where to search for help in order to solve them.
3. The students' skills must be sustainable and robust such that they can employ these skills later in their professional lives, i.e., they can identify and suitably approach moral issues even many years after attending the course and without studying ethics again.

Ethics for Nerds tries to achieve this by tailoring to the specific needs of computer science students, by keeping the course up-to-date and close to real life, and by making the course as motivating to students as possible (e.g., by using visually appealing course materials) such that they productively engage with the course content.

The course usually has four parts: The first and second part are dedicated to the essentials of philosophy that are needed throughout the course, i.e., ethics and argumentation theory. This includes the discussion of different moral theories (along with their problems) like consequentialism[8], deontological theories – such as Kant[9] or Scanlon's contractualism[10] –, and virtue ethics[11]. The students are also introduced to formal argumentation theory, such that they, e.g., gain proficiency in writing deductively valid arguments, are able to reconstruct arguments from texts, apply the Principle of Charity, spot fallacious reasoning and engage in structured discourse. Furthermore, students are introduced to important codes of ethics (especially the three codes referred to above) as well as the relationship of Codes of Ethics to moral theories. In the third part of the course, the skills acquired in the first two parts are applied. That part covers current topics related to computer science which bear moral relevance, like surveillance, machine learning, filter algorithms, machine explainability and perspicuity[12] and much more. In a short, final part, students are invited to look at problems that might become relevant in the future, such as robot and machine ethics, and at 'sci-fi' topics like superintelligences, which students usually have a great interest in.

Ethics for Nerds teaches this content with several main didactic elements:

- **Lecture**: In a weekly, 90-minute lecture (15 lectures in total), the students are introduced to new course content. The students are given access to the lecture slides such that they can use it for future reference. During the Covid-19 pandemic, high-quality videos were produced as substitutes for in-person lectures. For this purpose, the lectures were broken down into smaller, thematically self-contained parts of varying lengths (typically 10 to 25 minutes). Typical follow-up questions had been anticipated upfront and were incorporated into the videos based on experience from past iterations. This resulted in a total of 58 videos with a runtime of over 18 hours.
- **Training**: There are extensive training exercises that students are encouraged to solve in groups. While these exercises are completely voluntary, it is nevertheless recommended to engage with them. They come along with sample solutions so that it is easier for students to assess the quality of

[8] See Sinnott-Armstrong, "Consequentialism" in: Edward N. Zalta (ed.), *The Stanford Encyclopedia of Philosophy* (Fall 2021), available at https://plato.stanford.edu/archives/fall2021/entries/consequentialism/.

[9] See Kant, "Groundwork for the Metaphysics of Morals," 2019 [1785].

[10] See Scanlon, "What We Owe to Each Other," 1998.

[11] See Hursthouse/Pettigrove, "Virtue Ethics" in: Edward N. Zalta (ed.). *The Stanford Encyclopedia of Philosophy* (Winter 2018), available at https://plato.stanford.edu/archives/win2018/entries/ethics-virtue/.

[12] For an overview of various perspicuity properties – like transparency, traceability, explainability, and many more – and their significance for social issues of various kinds, see Sterz et al., "Towards Perspicuity Requirements," *2021 IEEE 29th International Requirements Engineering Conference Workshops* (REW) [2021], available at https://ieeexplore.ieee.org/document/9582408

their own answers and to show them an example of a complete solution to a task, which can also help them in preparing for the exam. Students are invited to talk about their solutions in office hours or in an online forum that all students are automatically granted access to.

- **Assignments**: In the computer science study program that *Ethics for Nerds* belongs to, it is common practice that exam admissions have to be earned by completing assignments during the semester. This also is the case in *Ethics for Nerds*. The mandatory assignments force students to engage with the course content in a timely manner and to practice the most relevant skills. They also enable students to get feedback on their performance from staff members and help the lecturers to assess how well the course is doing.
- **Discussions**: Participants are encouraged to discuss both with fellow students as well as with staff members. This is done in mandatory or voluntary discussion groups, in the lecture, the office hours, or in so-called *Nerd Nights*, which are informal get-togethers in the evenings, where students can engage with each other and with staff members, socialize, speak about the course content and their assignments, and eat pizza.
- **Term paper**: At the end of the course, participants have to write a term paper in which they take a stance on a topic from a moral perspective, come up with a convincing argument that is presented both formally and informally, and preemptively defend this argument against possible attacks. They can choose one of two to three topics with current relevance given out by the lecturers.
- **Exam**: The grade is mostly determined in an exam in which the students have to prove that they acquired the basic skills that were taught throughout the course.

Challenges

Ethics for Nerds presents several unusual didactic and practical challenges, four of which we are going to discuss here.

Firstly, students of computer scientists bring a unique set of skills that ethics lecturers should accommodate for. It is not recommendable to just take an ethics course from a philosophy curriculum and deliver it to computer scientists. Computer scientists are pre-educated in a way that philosophers are not. For example, they are usually better trained in formal logics, are skilled in algorithmic thinking, and have a more hands-on attitude towards solving problems that sometimes can be helpful but sometimes also turn out too hasty when applied to philosophical problems. All of that should be utilized and accommodated for in a lecture addressing computer science students.

Secondly, the high interdisciplinarity of Ethics for Nerds makes it necessary to build bridges between computer science and philosophy, which is not always an easy task. Some students think that philosophy is nothing more than exchanging unsubstantiated opinions. Furthermore, they worry that philosophers do not really understand the work of a computer scientist. Lecturers should preferably have expertise in both computer science and philosophy to ensure that an ethics course meets the needs of computer scientists.[13]

Thirdly, and most importantly, to fulfil its goals, *Ethics for Nerds* has to install a sustainable and robust set of skills in students that ideally does not need any revision or further immersion in their later professional life to stay intact and useful. Students of computer science need a pragmatic approach that makes the content of the course easily applicable to future real-life problems, as they are unlikely to ever enter moral education again.

[13] The general question of the appropriate profile of lecturers to teach a computer ethics or computer and society course is a longstanding contested issue. For a classic source, see Johnson, "Who should teach computer ethics and computers & society?" SIGCAS Computer and Society [Jun. 1994] 24 (2), pp. 6 – 13, available at https://dl.acm.org/doi/10.1145/181900.181901.

This also gives rise to a fourth challenge: while the course should be as student-friendly, up-to-date, didactically elaborate, engaging, interactive, and interesting as possible, it also must be feasible to teach it to several hundred participants each year. This makes it necessary to find a scalable way to achieve all the course's goals.

Best Practices

After six years of teaching and continually improving *Ethics for Nerds*, we have acquired some experience that seems worth sharing. In the following, we will describe some of the best practices of which we think that they are important to the success of *Ethics for Nerds*.

Tailoring to computer science students

As mentioned already, it is key to tailor towards the skills, interests, and expectations of the target audience. This makes it necessary to critically assess which philosophical concepts are relevant to students of computer science and which are not. This also includes simplifying philosophical theory where useful, while making transparent to the students where omissions and simplifications were made. Early on, we also saw that having students read philosophical texts – as it often is done in ethics courses – is often unnecessary and even counterproductive. The usefulness of original texts for computer science students is oftentimes very low compared to investing time in, e.g., doing training exercises or studying relevant real-life examples. We also found that students of computer science are mostly very capable of understanding even complex philosophical ideas and do not need to be sheltered from complicated thoughts as long as those are useful and practically relevant.

In order to properly select, edit, illustrate, and teach philosophical thoughts, a lecturer with a robust background in philosophy is needed. At the same time, the lecturer needs to be able to understand the perspectives, expectations, thoughts, and problems of computer science students, for which background knowledge in computer science is a prerequisite. Since teaching a course like *Ethics for Nerds* needs much deeper insights in philosophy than in computer science, we think that a philosopher with additional knowledge of computer science will on average be a better fit than a computer scientist with some limited background in philosophy.

Include argumentation theory

Whenever people who are not trained ethicists talk about ethics, things tend to get a bit 'hand-wavy'. While this is natural and understandable, it is not desirable. Ethical analyses of real-life problems originating in computer science can get very complicated at times. For this, it is very helpful to have formal tools to conceive arguments precisely, especially for students of computer science who are already trained in logics. For this reason, *Ethics for Nerds* introduces students to argumentation theory (or what is commonly called "Critical Thinking") with a particular emphasis on logical validity, the structure of a proper discourse and the Principle of Charity.[14] We found that this improves students' argumentation skills in general and helps them to pinpoint moral issues better. In the following, we briefly outline the significance of these particularly central aspects of Critical Thinking.

We put great emphasize on deductive, logical validity as well as the presentation of arguments in normal form, i.e., in a table where all premises and conclusions as well as each inference are made explicit. This is helpful when discussing ethics in at least two ways: first, it makes the structure of an argument very clear and thereby forces students to put thought into how the different premises in their moral reasoning relate. This helps in avoiding certain fallacies as well as gaps in their argument that would be hard to bridge. Secondly, it forces

[14] See Blackburn, "The Oxford Dictionary of Philosophy". Oxford University Press, p. 62.

students to make each of their premises explicit. Thereby, they can avoid implausible implicit premises that are commonly underlying moral reasoning by non-experts.

Furthermore, we go into the theory of a proper discussion, e.g., how to attack an argument in sound and systematic ways as well as how to anticipate and correctly counter such attacks. We emphasize that discussion is about finding truth and not about winning. Experience shows that especially discussions about morals and ethics can get very heated, presumably because people usually identify heavily with their moral beliefs and judgements. We saw that knowing how to properly structure a discourse helps students in engaging in more productive discussions, even if they do not spell out each of their arguments formally.

A productive discussion is usually not possible without one main principle that we put strong emphasis on: the already mentioned Principle of Charity. It says that arguments by others should always be taken with good will and that efforts should be made to understand it in its strongest sense. By employing it, discussion is less likely to digress to unimportant points and more likely to lead the discussing parties to discovering truth.

Tailor towards skills rather than knowledge

It is common practice in philosophy and ethics to focus on the original theories, views, and arguments from prominent scholars like Kant, Foot, or Bentham. For the most part, however, these are primarily academic contributions that oftentimes do not help in real-life ethical decision making out of the box. There is no point in teaching just theories that students will learn for the exam and will forget soon after if the goal of the course is to equip them with the necessary skills to make morally informed decisions in their professional lives. To this end, a lecturer should design the course with the skills in mind that they want students to acquire and carefully select the course content accordingly. This includes making transparent to students which skills are relevant, giving students plenty of opportunities to train these skills and to get feedback on their results, and also to test for these skills as much as possible in assignments and exams. We found it useful to allow students to bring a cheat sheet to the exam or even to write open-book exams, so as to encourage students to sharpen the skill set they acquired throughout the lecture instead of just learning facts by heart.

For us, the set of skills that we want the students to acquire includes (but is not limited to) identifying potential moral problems, reasoning within certain moral frameworks, analyzing situations in respect of what plausibly is the morally right thing to do, writing logically valid arguments and engaging in productive, charitable discussions. Among the things that we found *not* to be useful were: knowing the edifice of ideas by famous philosophers in detail, reading and understanding original philosophical texts, and learning about philosophical history.

Also, we had positive experiences with being very open on the results of students' works, while being strict about the method. If, for example, students should come up with a logically valid and reasonably plausible argument for or against a given claim, we did not grade them based on whether they made a pro or a contra case, but only based on how well they crafted their arguments – usually targeting deductively valid, informative, and well-presented arguments with plausible premises that the student gave adequate reasons for.

Work with up-to-date examples and practically relevant topics

For students to become equipped for their professional lives and to retain interest in the lecture, it proved very useful to base especially the third and fourth part of the lecture on real-life, up-to-date examples. This makes it necessary to continually update the course. Based on our experiences in the Covid-19 pandemic during which we had to focus on creating videos and overcoming the challenges of online teaching, we found that the course does reasonably well without major updates for two to three years, but after this time it should be adapted to new developments at the latest.

Encourage discussions

In our experience, students greatly profit from discussions – both with each other and with the team. Encouraging discussions can be a tedious task for lecturers since students often do not see the immediate advantage of discussions and sometimes are a bit reluctant to discuss with classmates they do not know well. The best way to foster discussion greatly depends on the setup of the course and the number of students attending it. For smaller courses, we found tutorials with few students at a time very helpful, while for larger courses we made good experiences with making the submission of a number of discussion protocols mandatory for the exam admission. In a discussion protocol, a group of students has to write down the main results of a discussion on a topic of their choice on not more than one page. The goal of that is not to make sure that the result of the discussion is correct, but to get the participants into actually talking with each other. In general, we found that there unfortunately is no one-size-fits-all approach. Therefore, it is important to provide students with plenty of opportunities to engage with each other and to frequently prompt them to do so. In our experience, making some discussion mandatory helps in bringing students together and provides them with opportunities to engage in discussions that are both useful and enjoyable.

Make the course fun and visually appealing

In our experience with *Ethics for Nerds*, a learning-friendly and visually appealing presentation, especially on the lecture slides, helps students a lot to stay motivated and to understand the course content better. We aim to avoid slides with just a wall of text on them and rather try to convey each thought with a figure, a diagram, a schematic or the like. Especially with content outside the student's own subject area, we found it important to make sure that as little effort as possible is needed to grasp each key idea.

Conclusion

Ethics for Nerds is a lecture that is well received by the students and that retains a high popularity at Saarland University. Among others, it was awarded a competitive national award by the largest private science foundation in Germany, the *Stifterverband*. Courses like it are needed from a consequentialist perspective, to ensure adequate responsibilities, and from a perspective of professional ethics, especially regarding different codes of ethics and conduct. We hope that many more ethics courses are included in computer science programs. Given how important these can be, it is paramount that lecturers from different universities share their experiences with each other and thus ensure that teaching concepts grow in quality and scale better.[15] Because ethics courses in computer science study programs promote the public good, they deserve adequate financial resources to provide not just a quick overview of academic ethics but instead a solid foundation for moral decision making.

Acknowledgements

This research has received financial support by the *DFG* grant 389792660 as part of *TRR* 248 (*Center for Perspicuous Computing*), and by the Volkswagen Foundation grants AZ 98514 (*Explainable Intelligent Systems*).

[15] The Erasmus+ transnational project *Ethics4EU* offer a promising opportunity for building such a network, see http://ethics4eu.eu/.

References

Askel, Amanda. "The Moral Value of Information." EA Global: Boston 2017. Science Center, Harvard University, 2. – 4. Jun. 2017, Harvard Science Center, Cambridge, USA. Talk available at https://www.youtube.com/watch?v=o8rVscSHJT4 and transcript available at https://www.effectivealtruism.org/articles/the-moral-value-of-information-amanda-askell/.

Baum, Kevin, et al. "From Responsibility to Reason-Giving Explainable Artificial Intelligence." Philosophy & Technology, forthcoming.

Blackburn, Simon. "The Oxford Dictionary of Philosophy." Oxford University Press, 2008.

Johnson, Deborah. "Who Should Teach Computer Ethics and Computers & Society?" SIGCAS Computer and Society [Jun. 1994] 24 (2), pp. 6 – 13. Available at https://dl.acm.org/doi/10.1145/181900.181901.

Kant, Immanuel, et al. "Groundwork for the Metaphysics of Morals." Oxford University Press, 2019 [1785].

Gotterbarn, Donald. "Informatics and professional responsibility." Science and Engineering Ethics [2001] 7, pp.221–230. Available at https://doi.org/10.1007/s11948-001-0043-5.

Howard, Ronald A. "Information Value Theory." IEEE Transactions on Systems Science and Cybernetics [Aug. 1966] 2 (1), pp. 22-26. Available at https://ieeexplore.ieee.org/abstract/document/4082064.

Hursthouse, Rosalind/Pettigrove, Glen. "Virtue Ethics" in: Edward N. Zalta (ed.). The Stanford Encyclopedia of Philosophy (Winter 2018). Available at https://plato.stanford.edu/archives/win2018/entries/ethics-virtue/.

Noorman, Merel. "Computing and Moral Responsibility" in: Edward N. Zalta (ed.). The Stanford Encyclopedia of Philosophy (Spring 2020). Available at https://plato.stanford.edu/archives/spr2020/entries/computing-responsibility/.

Scanlon, T.M. "What We Owe to Each Other." Cambridge, MA: Harvard University Press, 1998.

Sinnott-Armstrong, Walter. "Consequentialism" in: Edward N. Zalta (ed.). The Stanford Encyclopedia of Philosophy (Fall 2021). Available at https://plato.stanford.edu/archives/fall2021/entries/consequentialism/.

Sterz, Sarah, et al. "Towards Perspicuity Requirements." 2021 IEEE 29th International Requirements Engineering Conference Workshops (REW). IEEE, 2021. Available at https://ieeexplore.ieee.org/document/9582408

Talbert, Matthew. "Moral Responsibility" in: Edward N. Zalta (ed.). The Stanford Encyclopedia of Philosophy (Winter 2019). Available at https://plato.stanford.edu/archives/win2019/entries/moral-responsibility/.