Towards teaching Computer Ethics

C de Ridder \textsuperscript{a}  L Pretorius \textsuperscript{b}  A Barnard \textsuperscript{c}

\textsuperscript{a} Unisa, South Africa, driddc@unisa.ac.za
\textsuperscript{b} Unisa, South Africa, pretol@unisa.ac.za
\textsuperscript{c} Unisa, South Africa, barnaa@unisa.ac.za

Abstract: With the advent of the Information Age and global connectivity, the ethical use of Information and Communication Technologies (ICT) has become essential. As the drive towards the establishment of a so-called IT profession gains momentum, ethical conduct and codes of ethics have recently been formulated and introduced formally. Forerunners in this initiative are the ACM and the IEEE. Of particular significance is the ImpactCS Project commissioned by the joint taskforce of the ACM and IEEE, and funded by the USA National Science Foundation. The increased globalisation and inherent nature of ICT, transcend physical and cultural borders, making it increasingly difficult to enforce accepted laws, regulations, and codes of conduct. It is thus the responsibility of Computer Science and Information Systems instructors to teach and instill values and ethics in each and every student. Therefore, we investigate some issues pertaining to the teaching of computer ethics. We conclude this paper by presenting a possible model to be used in the teaching of computer ethics, and apply this model to our own institution.

Keywords: Computer ethics, curriculum studies, computer science education
Computing Review Categories: K.3.2, K.4.1, K.7.4

1. Introduction

The terms “ethics”, and in particular “computer ethics” (CE), may seem vague and not at all important to many computer scientists, especially those who hold the view that technologists should deal with technology, and philosophers with philosophy and ethics [6]. It is therefore perfectly reasonable to expect that the teaching of CE may also be considered by some as of little importance. Yet many CS/IS instructors hold a different view, namely that the teaching of CE to CS/IS students is as important as the teaching of technological topics. In this paper we explore this topic in some detail, in particular what CE is, and why and how it may be taught.

We discuss a generic model based on the ImpactCS proposal that may be used when considering the teaching of CE, and then apply it to our own (distance education) institution. In this paper we restrict our attention to the ACM/IEEE approach to the teaching of CE.

2. Computer ethics

Many authors [1, 6, 11, 20, 21, 27] who discuss the topic of ethics simply assume that their audience is conversant with the notion of ethics. We find it useful to state clearly what we consider this broad term to imply, and its particular relevance to CE. The literature offers numerous definitions and notions of the terms ethics and computer ethics [15], but for the purposes of this discussion we concur with Britz [4] who attempts to define ethics as follows:

\textit{The ethical actions of a person can be described in general terms as those actions which fall within the range of those activities that would be regarded as “good”. It relates this to the question of what is good and bad in terms of human actions.}

and Baase [2] who simply says:

\textit{Ethics is the study of what it means to “do the right thing.”}

These definitions may seem somewhat non-scientific. Surely that which one individual may understand under the term “good”, may differ from that of another individual. The Odhams Dictionary of the English Language defines the term “good” as \textit{a merit, a moral quality, a virtue, an advantage, benefit, profit}. However, what appeals to us here is the strong emphasis on actions, and the practical nature of ethics. This conforms to Aristotle’s classification of different kinds of science, viz theoretical, productive and practical science [18], with ethics, politics, and economics constituting the practical sciences. This is but one of the reasons why we consider the teaching of CE to form an integral part of the training of any IT professional. Again, Baase [2], supports this by stating that computer ethics involves “ethical issues faced by a computer professional as part of the job”. We therefore describe the term computer ethics (CE), as \textit{the study of those behavioural actions}...
Numerous ethical and social issues caused by computer technology arise. According to Martin and Yale-Weltz [17] these issues have three unique characteristics:
- new concerns are rapidly emerging,
- computer ethics presents a continuous stream of new situations, and
- computerised information systems are usually complex.

Ideally students should be equipped with theories of philosophy and ethics pertinent to these characteristics, as well as the skills to analyse, evaluate, and react appropriately to ethical dilemmas which may arise during their careers as IT professionals. Forester and Morrison [9] identified the following main categories (or groupings of topics) in which these ethical and social concerns usually arise:
- computer crime and computer security,
- software theft and intellectual property rights,
- computer hacking and the creation of viruses,
- computer and information system failure,
- invasion of privacy,
- social implications of artificial intelligence and expert systems, and
- workplace computerisation.

It is often within the context of these categories, that the behavioural actions of the IT professional impact on society. We believe this to be one of the main reasons as to why CS/IS instructors should sensitise their students, the IT professionals of the future, to actions that underpin concepts such as “good”, “moral”, “ethical”, and “beneficial” for society at large. This is in agreement with Searls [24] who states that the teaching of CE should aim towards several important goals, viz.:
- increased sensitivity to ethical concerns and situations, as well as
- reasoning about alternative courses of action and the integrity to make moral decisions.

A certain historical perspective is gained by recalling three major contributions in the teaching of computer science, viz. the development of Computing Curricula 1991 (CC91) by a joint task force of the ACM and the IEEE Computer Society as a framework for the previous iteration of the computer science curriculum, the subsequent ImpactCS Project, and the Computing Curricula 2001 (CC2001) for the current iteration of the computer science curriculum.

The nine core subject areas in CC91 were algorithms and data structures, architecture, artificial intelligence and robotics, database and information retrieval, human-computer communication, numerical and symbolic computations, operating systems, programming languages, and software methodology and engineering. The recognition of the social, ethical and professional context of computer science was included as one of the foundational principles, but CC91 fell short of providing sufficient detail and guidelines for implementation of CE within the curriculum.

In 1994 the subsequent ImpactCS Project was funded by the US National Research Foundation to produce a more rigorous definition of the content area of ethics and social impact within computer science, and a methodology for integrating these topics across the computer science curriculum.

In 1998 another major review of curriculum guidelines for undergraduate programs in computing, Computing Curricula 2001 (CC2001), was undertaken [33]. Fourteen knowledge areas were identified, including one entitled Social and Professional Issues. CC2001, Draft (February 1, 2001) [33] contains extensive recommendations for all the knowledge areas except for Social and Professional Issues. This task still remains to be completed. Since we are of the opinion that the introduction of CE into our CS/IS programs is a matter of some urgency, we decided to use the ImpactCS Project framework as the present guideline. When the CC2001 recommendations on CE become available, we shall study and consider them for incorporation.

4. Who should teach computer ethics?

Computer ethics is by definition a subject that involves multiple disciplines, certainly at least computer science and philosophy. When considering the teaching of CE an obvious question arises: Should it be taught by philosophers, or should computer scientists take this task upon themselves?
The vast and growing literature on CE distinguishes between at least two main schools of thought. An argument in favour of philosophers is based on the view that the goals of such a course should determine who ought to teach it. According to Johnson [14] these goals are:

1) to make students (future computing professionals) aware of the ethical issues surrounding computers,
2) to sensitize them to ethical issues in the use of computers, and in the practice of the computing professions,
3) to convey to students a deep understanding of the ways in which computers change and impact on society, and
4) to provide conceptual tools and develop analytic skills for ethical decision making within the computing world.

Johnson maintains that, while having to do with computers, these goals and issues are at base ethical, social, and professional, and that: nothing in the training of computer scientists and engineers prepares them for these types of activities. [... Philosophers are specifically trained to analyze issues, to uncover the assumptions or implications of arguments and claims, and to develop arguments. Philosophers trained in ethical theory have a repertoire of concepts and frameworks that are useful for examining situations, identifying the moral core, and evaluating and justifying courses of action [14].

She therefore envisages the philosopher as the teacher, assisted by computer scientists when and where required.

While the goals of Johnson are readily recognized, her conclusions do not enjoy equal acceptance. Indeed, a general view is that computer scientists are capable of learning the ethical theories and strategies. Given some training in ethical issues, provided by books, workshops, seminars, etc., and assisted by philosophers when necessary, they would be able to effectively include computer ethics in their curricula (Gotterbarn as quoted in [14]). Moreover, the fundamental part that social and ethical issues should play in modern computer science would be best emphasized to students by the involvement of senior computer science faculty in the planning, implementation and teaching thereof (Martin as quoted in [14]). We anticipate that once computer ethics has been established as main stream computer science, it will be taught and researched like all other fields and that the question under discussion will simply disappear. Having established that computer scientists should be responsible for the teaching of CE, we now focus on the framework for our model, viz. the knowledge units proposed in [13, 16, and 17].

5. The ImpactCS framework

The ImpactCS Project proposes five knowledge units [16, 17] as the basis for CE courseware. These knowledge units and its associated topics are:

Unit 1 - Responsibility of the Computer Professional
1.a) history, development, and impact of computer technology,
1.b) why be ethical,
1.c) major ethical models,
1.d) definition of computing as profession, and
1.e) codes of ethics and professional responsibility for computer professionals (in [6, 23] the reader is presented with useful information as to the ACM and IEEE codes of conduct).

Unit 2 - Basic elements of ethical analysis
2.a) ethical claims can and should be discussed rationally,
2.b) ethical choices cannot be avoided, and
2.c) easy ethical approaches and solutions are questionable, i.e. it is hard.

Unit 3 - Basic skills of ethical analysis
3.a) arguing from example, analogy, and counter-example,
3.b) identification of stakeholders in concrete situations,
3.c) identification of ethical issues in concrete situations,
3.d) application of ethical codes to concrete situations, and
3.e) identification and evaluation of possible courses of action.

Unit 4 - Basic elements of social analysis
4.a) social context influences the development and use of technology,
4.b) power relations are central in all social interactions,
4.c) technology embodies the values of the developers,
4.d) populations are diverse, and
4.e) empirical data are crucial to the design and development processes.
5. **Basic skills of social analysis**

5.a) identification and interpretation of the social contexts of a particular implementation,

5.b) identification of assumptions and values embedded in a particular system, and

5.c) evaluation by means of empirical data of a particular implementation of a technology.

Applying this framework to one’s own institution, faculty, and department, as well as the formulation of an appropriate educational model, now needs to be considered.

6. **Towards a model for teaching computer ethics**

6.1 Integrated VS stand-alone approach

The ImpactCS Project [17] identified two main strategies for the teaching of CE. These are:

- the integration of the five knowledge units in the existing material on each level, and
- a stand-alone course dedicated to CE material.

However, it favours the teaching of CE in an integrated fashion [13, 16, 17, 19, 21]. According to [17], the key to the integrated approach lies in the overall coordination of the CE curriculum. One example of this approach is presented in [30] where Yale-Weltz describes a faculty initiative at the Seattle Pacific University for the teaching of CE at various levels. Although ImpactCS prefers the integrated approach, it is noteworthy that most literature references [3, 6, 11, 22, 25, 29] deal with the practical implementation from the point of view of a stand-alone course. A possible reason for this phenomenon could be that although CS/IS instructors are in agreement with the principle of teaching CE in an integrated fashion, the practical implementation thereof is problematic.

Various examples of stand-alone courses are presented in the literature. In [22] Roberts presents a curriculum as well as assignments to be completed by students, ranging from written, oral, to on-line participation for his course. Wahl [29] presents yet another stand-alone course curriculum with reference to Project ImpactCS, ACM, and IEEE professional codes of conduct. Other examples may be found in [3, 6, 11, 25].

6.2 Building the generic model

While we in principle support the integrated approach suggested by ImpactCS, it is clear from the literature [3, 6, 11, 22, 25, 28, 29] that in most cases, departmental pragmatics and logistics necessitate a combination of the integrated and stand-alone approach. Since our goal is to make all our students aware of the issues in CE, and to equip them with the capabilities and skills of evaluation, decision-making, and appropriate responsible action, we focus our attention on core modules as the vehicle for teaching CE. By core modules we mean those modules that are compulsory in all our degree programs. Our combined approach will facilitate an early introduction (on first level), continued discussion at following levels, and the integration of CE topics in the core modules. Together with our subsequent stand-alone third level module we expect to achieve maximum coverage with minimum overlap [17].

We propose a generic model for developing a CE curriculum, based on the topics associated with the knowledge units of section 5. A simple algorithm for mapping these topics to undergraduate modules in a coordinated way, is presented. This algorithm aims at achieving maximum coverage with minimum overlap. It may be applied at any institution regardless of the medium of instruction, but in section 7 we investigate its application in our distance education environment. In the algorithm we use set-theoretic notation and pseudo-code for clarity and conciseness. The topics defined by the ImpactCS Project (see section 5) are represented as the universal set $T$:

**Algorithm to match these topics with the undergraduate core module structure**

**Step 1.** Identify suitable first and second level core modules. Let $m$ denote the number of such modules.

**Step 2.** Match applicable topics from $T$ to existing module outcomes. Let these topics be the set $A$.

**Step 3.** Identify the topics not represented thus far as $N = T - A$.

**Step 4.** For each core module $i$, identify those topics from $N$ which may be covered, and can be integrated into the core module. Call this set $C_i$.

**Step 5.** The pair-wise set-theoretic intersection of the sets $C_i$ will determine overlap of topics identified in step 4:
for $i = 1, \ldots, m$
for $k = i+1, \ldots, m$

determine $C_i \cap C_k$

do until $C_i \cap C_k = \emptyset$
oreach $x \in C_i \cap C_k$ determine the pair-wise best match and modify $C_i$ and $C_k$ accordingly, i.e. if best match dictates $x \in C_p$, then let $C_q = C_p \cap \{x\}$ where $p, q \in \{i, k\}$ and $p \neq q$.

do

determine the pair-wise best match and modify $C_i$ and $C_k$ accordingly,

end do,
end for $k$,
end for $i$.

Step 6: Indicate the union of these modified sets $C_i$ by $C$.

Step 7: The remaining topics, $T - (A \cup C)$, should constitute the basis of the stand-alone module.

It is clear that this algorithm presents a combined (i.e. combination of the integrated and stand-alone) approach which guarantees meaningful integration of appropriate topics across the core modules of the CS/IS curriculum. Indeed, the stand-alone module is intended to cover the remaining topics (step 7). In determining the best matches the paper of Yale-Weltz [30] proved particularly useful (step 5).

7. Applying the model at our institution

Our own institution is a distance education one. Unisa, established in 1873, is the oldest and largest university in South Africa. It is one of 11 mega distance education universities (more than 100 000 students) in the world. Its Department of Computer Science and Information Systems, with a teaching staff of over 40, has 11 000 students, enrolled for 29 000 module papers - arguably the largest CS/IS department in the country! Indeed, teaching CE to thousands of students by means of distance education is certainly a unique but worthwhile challenge.

7.1 A team approach in teaching computer ethics

The Unisa BSc (three year) degree programmes have a modular structure, with fourteen modules at first level, eight at second level, and eight at third level. At Unisa the teaching is usually done in small teams, with particular emphasis on planning, coordination and student support. In our CS/IS department each module is offered by a team consisting of two to four lecturers, who are collectively responsible for the planning, implementation and teaching of their module. Each individual is typically involved in the teaching of two undergraduate modules, with student numbers per undergraduate module ranging from 100 to 1600 or more. This team approach turns out to be ideally suited to the teaching of CE, using our combined approach. The CE team will amongst other be responsible to:

1. Apply the algorithm above to determine the core modules on first and second level and to identify the topics from $T$ already covered in these core modules, set $A$ above.

2. The CE team should furthermore identify which of the remaining topics ($N = T - A$) can be addressed sensibly by the existing module outcomes, set $C$ in the algorithm, and they should then ensure that these topics be introduced to the existing module structure so that the module outcomes match the topics in set $C$.

3. Lastly the CE team should design and teach a third level module that addresses all of the topics not covered by the sets $A$ and $C$.

The advantages of this dedicated team based approach to the teaching of CE include the following:

- Lecturing staff involved with the instruction of the core modules will not be burdened with developing and teaching the CE topics as the CE team will be responsible for this.

- It guarantees the structured and co-ordinated teaching of ethical issues and concerns embodied by the topics, ensuring minimum overlap yet allowing for maximum coverage of ethical issues.

7.2 The status quo at Unisa

When embarking on such a project or investigation, it makes sense to take note of the status quo. So, let us briefly examine what is offered in terms of the teaching of CE in our courseware at Unisa at present.

While many of the text books and study material used by our department may in passing refer to aspects of CE, we only list those modules where it assumes a somewhat more central role. For each module we also list the relevant topics addressed in brackets:

CEM101 - End User Computing [31]

Topics: Security, copyright and the law, treating issues such as understanding software copyright and security questions, awareness of privacy
issues, data protection, viruses and anti-virus measures, what happens in the case of a power failure. \{1.a, 1.b, 4.a\}

COS113 - Computer Systems - Fundamental Concepts \[26\]

Topic: Netiquette - focussing on the unwritten rules for acceptable conduct when using the internet. \{None\}

INF120 - Human Computer Interaction \[32\]

Topics: Netiquette, security of digital data, safety and the repercussions of computer hardware and software failure, the provocative question concerning human failure (which may include unethical conduct) is also posed. \{1.a, 1.b, 3.b, 4.a\}

We observe:

- The modules in question are all first level modules.
- Even at this basic level the overlap is significant.
- CEM101 and COS113 are core modules in the BCom Informatics and the BSc degree programs respectively. INF120 is an elective module.
- The positive side however is that students are sensitised at an early stage in their studies, and the spreading of the CE material across a number of modules emphasises the importance thereof.
- The negative side is that CE is neglected at the second and later levels. Many of the key CE issues and topics are not addressed.
- The topics 1.a, 1.b, 3.b, and 4.a are addressed in these modules. However not in an integrated manner, with the bulk of these topics dealt with in INF120, an elective.

7.3 Applying the algorithm

We apply the algorithm to facilitate our integration of CE in the computer science programs of the department (a similar iteration can be performed for the IS programs):

7.3.1 Step 1 of the algorithm, the core modules:

First of all we identify the following core modules:
- COS113: Computer systems - fundamental concepts,
- COS112: Programming: practical.

7.3.2 Step 2 of the algorithm, set A:

From the previous section, \(A = \{\}\), the empty set.

7.3.3 Step 3 of the algorithm, the set \(N\):

\(N = T\)

Set \(N\) thus consists of all the topics.

7.3.4 Step 4 of the algorithm, the sets \(C_j\):

Two programming core modules have been identified as modules which allow the inclusion of certain aspects of CE. We briefly mention the CE topics which could be incorporated in these modules - the associated ImpactCS topics are indicated in brackets.

COS113:
- History and evolution of computers and their impact on society (1.a, 4.a, 4.b).
- Virus protection, security and safety issues (1.a, 3.b, 3.c, 3.e).
- Internet-related security and privacy issues (3.c, 3.e, 4.a).
Set \(C_1 = \{1.a, 3.b, 3.c, 3.e, 4.a, 4.b\}\).

COS112:
- The history and evolution of computers and their impact on society (1.a, 4.a, 4.b).
- Copyright and virus protection (3.b, 3.c, 3.e).
Set \(C_2 = \{1.a, 3.b, 3.c, 3.e, 4.a, 4.b\}\).

COS 211:
- Ethical dilemmas should be introduced to students, with an emphasis on situations which presents difficult choices, sometimes even no “good” option (2.a, 2.b, 2.c).
- Arguing from example, analogy, and counterexample (3.a).
Set \(C_3 = \{2.a, 2.b, 2.c, 3.a\}\).

COS212:
- Ethical models should be studied informally, and rational decision making should be emphasized (1.c).
- ACM and IEEE codes of conduct should be introduced (3.d).
- Students must be sensitised to ethical issues related to program design and should be provided with at least one programming assignment in which they should identify some of the ethical and social issues pertaining to the specific program (3.c). Riser and Gotterbarn [11, 21] provide applicable examples.
Set \(C_4 = \{1.c, 3.c, 3.d\}\).

7.3.5 The pair-wise disjoint intersection of the sets \(C_j\):
Below we only give the final result of step 5 of the algorithm, and note that in determining best matches, [30] provides useful information:

COS113:
- History and evolution of computers and their impact on society (1.a, 4.a, 4.b).
- Internet-related security and privacy issues (4.a).
The modified set $C_1 = \{1.a, 4.a, 4.b\}$.

COS112:
- Copyright and virus protection (3.b, 3.e).
The modified set $C_2 = \{3.b, 3.e\}$.

COS 211:
- Ethical dilemmas should be introduced to students, with an emphasis on situations which present difficult choices, sometimes even no “good” option (2.a, 2.b, 2.c).
- Arguing from example, analogy, and counterexample (3.a).
The modified set $C_3 = \{2.a, 2.b, 2.c, 3.a\}$.

COS212:
- Ethical models should be studied informally, and rational decision making should be emphasized (1.c).
- ACM and IEEE codes of conduct should be introduced (3.d).
- Students must be sensitised to ethical issues related to program design and should be provided with at least one programming assignment in which they should identify some of the ethical and social issues pertaining to the specific program (3.c).
The modified set $C_4 = \{1.c, 3.c, 3.d\}$.

Therefore,
$$C = \{1.a, 1.c, 2.a, 2.b, 2.c, 3.a, 3.b, 3.c, 3.d, 3.e, 4.a, 4.b\}.$$  

7.3.6 The stand-alone module:
The remaining topics will constitute the basis of the stand-alone module. They are:
\{1.b, 1.d, 1.e, 4.c, 4.d, 4.e, 5.a, 5.b, 5.c\},
and may be divided into three groups. While the algorithm guarantees maximum coverage with minimum overlap, it remains the prerogative of the CE team to repeat certain topics for the purpose of deepening knowledge and insight. The three groups are as follows:

1. Generic CE-topics
- Formal discussion of major ethical models (1.c).
- Formal approach towards ethical decision-making (2.a, 2.c, 3.b).
- Professionalism (1.b, 1.d, 1.e).
- Codes of ethics, ACM and IEEE codes of conduct (1.e, 3.d).

2. Concepts concerning programming languages
- The impact of the designers’ values and assumptions (4.a, 4.c, 5.b).
- Exception handling (5.c).
- Security and safety issues (1.a, 3.d).
- Extension and ownership (1.a, 3.b).

3. Concepts concerning system design
- The developers responsibility (1.a, 4.e).
- The implications of the fact that systems are designed for real people of different social and cultural backgrounds (4.a, 4.d, 4.e, 5.a).
- File systems, security protection from hackers, viruses, worms and Trojan horses (1.a, 4.b).

Summarising, we presented a model that may be used to map the topics to module outcomes in a meaningful manner. We also applied this model to our own institution and showed how the algorithm provided us with a strategy for integrating the teaching of CE with our particular existing module structure. Our investigation resulted in a combination of the two approaches of section 6.1, [17], in a way which reflects our present departmental structure and distance mode of instruction.

8. Future initiatives

What remains to be done for the computer science curriculum is the development of the courseware and the investigation of various pedagogical strategies for teaching CE according to the schedule produced by our algorithm. A favoured pedagogical tool is case studies. Meaningful contributions on the use of case studies in the teaching of CE, are to be found in [3, 6, 11, 20, 22, 25]. The opportunities offered by modern communication technologies, including the world wide web and internet, need to be explored, [8, 10, 12].

In a similar way, CE may be introduced into our Information Systems curriculum.

While the ACM/IEEE approach to teaching CE as explored in this paper, may seem biased towards society, we plan to investigate other approaches in which the individual is afforded a somewhat more central role.
9. Conclusion
In conclusion we support the ACM/IEEE Joint Curriculum Taskforce’s position regarding the teaching of CE [17]:

Undergraduates need to understand the basic cultural, social, legal, and ethical issues inherent in the discipline of computing. They should understand where the discipline has been, where it is, and where it is heading. They should understand their individual roles in this process, as well as appreciate the philosophical questions, technical problems, and aesthetic values that play an important part in the development of the discipline. ... Students also need to develop the ability to ask serious questions about social impact and to evaluate proposed answers to those questions.

Computer Science and Information Systems curricula in South Africa however do not at present require the inclusion of CE in presentation of the subject matter. The situation seems to be similar to that reported by Clarke in 1992 [5]. A vast number of institutions in South Africa train computer professionals, equipping them with the required technical skills, without sufficient emphasis on the ethical and social implications of their actions which are irrevocably intertwined in the career-path of a computer professional. If we are to adequately prepare our students for their future careers as responsible IT professionals, and if we are to consider the accreditation initiatives that are currently underway in the USA, and elsewhere in the world, which will ensure that all future IT professionals are exposed to similar subject-related concepts, theories, and ethical concerns, we need to proceed with the teaching of CE in an integrated fashion.

10. References


[14] D. G. Johnson; Who should teach computer ethics and computers and society?,